NASA Conference Publication 3240

Space Technology Interdependency Group

Seventh Annual Workshop on Space Operations Applications and Research (SOAR '93)

Volume I

Proceedings of a workshop held in
Houston, Texas
August 3–5, 1993

(NASA-CP-3240-Vol-1) SEVENTH ANNUAL WORKSHOP ON SPACE OPERATIONS APPLICATIONS AND RESEARCH (SOAR 1993), VOLUME 1 (NASA. Johnson Space Center) 477 p

N94-34019
--THRU--
N94-34062
Unclas

H1/99 0000580

NASA Conference Publication 3240

Seventh Annual Workshop on Space Operations Applications and Research (SOAR '93)

Volume I

Kumar Krishen, Editor
NASA Lyndon B. Johnson Space Center
Houston, Texas

Proceedings of a workshop sponsored by the
National Aeronautics and Space Administration
Washington, D.C., and the U.S. Air Force, Washington, D.C.,
and held in
Houston, Texas
August 3-5, 1993

National Aeronautics and
Space Administration

CONTENTS

INTRODUCTION

WELCOME / OPENING ADDRESSES

PLENARY SESSION HIGHLIGHTS

PANEL DISCUSSION HIGHLIGHTS

KEYNOTE ADDRESSES

SECTION I: ROBOTICS AND TELEPRESENCE

Session R1: NAVIGATION, MACHINE PERCEPTION, AND EXPLORATION
Session Chair: Dr. Brian Wilcox

Microrover Research at JPL 1

Design of the MESUR/Pathfinder Microrover 2

Air Force Construction Automation/Robotics 3

Lunar Exploration Rover Program Developments 4

Session R2: ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES: PANEL PRESENTATIONS
Session Chair: Capt. Paul Whalen

Robotics and Telepresence (R&T) technology Challenges
Panelists:
Dr. Chuck Weisbin, NASA/JPL (In-Space Robotics Challenges)
Mr. Chuck Shoemaker, U.S. Army (Major Technology Challenges for DOD UGVprogram)
Dr. Harold Hawkins, U.S. Navy
Maj. Michael Leahy, Jr., USAF (Depot Telerobotics: The Challenges)
Mr. Joe Herndon, U.S. Dept. of Energy (Robotics and Telepresence Research Challenges)
Mr. Charlie Price, NASA/JSC (Space Robotics and Telepresence Research Challenges)

Robotics & Telepresence Research Challenges: Panel Presentation 13
Major Technology Challenges for DoD UGV program 1993-2000 16

PRECEDING PAGE BLANK NOT FILMED

Depot Telerobotics: The Challenges 22
Robotics and Telepresence Research Challenges – A Department of Energy Perspective 24
Space Robotics and Telepresence Research Challenges 32

Session R3: **ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES: PANEL DISCUSSION**
Session Chair: **Capt. Paul Whalen**

Panel Discussion on Robotics and Telepresence (R&T) Technology Challenges 39

Session R4: **REMOTE INTERACTION WITH SYNTHETIC ENVIRONMENTS**
Session Chair: **Dr. Harold Hawkins**

Shared Virtual Environments for Aerospace Training 49

Surgery Applications of Virtual Reality 50

A Study of Navigation in Virtual Space 51

RoboLab and Virtual Environments 61

Session R5: **REMOTE INTERACTION WITH PHYSICAL SYSTEMS**
Session Chair: **Mr. Joe Herndon**

Development and Demonstration of a Telerobotic Excavation System 69

A Teleoperated System for Remote Site Characterization 75

Vehicle Development of Lunar/Mars Exploration 85

Controlling Telerobots with Video Data and Compensating for Time-Delayed Video Using Omniview™ 86

Session R6: **MANIPULATORS AND END EFFECTORS**
Session Chair: **Mr. Charlie Price**

Dexterous End Effector Flight Demonstration 95

Undersea Applications of Dexterous Robotics 103

Robotic Technologies of the Flight Telerobotic Servicer (FTS) Including Fault Tolerance 112

EVA-SCRAM Operations 121

Hardware Interface for Isolation of Vibrations in Flexible Manipulators—
Development and Applications 130

Session R7: ROBOTIC OPERATIONS: SPACE TERRESTRIAL
Session Chair: Mr. Charles Shoemaker

Ground Vehicle Control at NIST: from Teleoperation to Autonomy 137

Intelligent Vehicle Control: Opportunities for Terrestrial-
Space System Integration 143

The Servicing Aid Tool: A Teleoperated Robotics System for
Space Applications 144

Robotic Vehicle Mobility and Task Performance – A Flexible Control Modality
for Manned Systems 156

SECTION II: AUTOMATION AND INTELLIGENT SYSTEMS

Session A1: ARTIFICIAL INTELLIGENCE I
Session Chair: Dr. Peter Friedland

Artificial Intelligence at AFOSR 157

The Stanford How Things Work Project 167

Acting to Gain Information: Real-time Reasoning Meets
Real-time Perception 177

From Conditional Oughts to Qualitative Decision Theory 178

Finding Accurate Frontiers: A Knowledge-Intensive Approach
to Relational Learning 187

Reinforcement Learning in Scheduling 194

Constraint-Based Integration of Planning and Scheduling for Space-Based
Observatory Management 200

Session A2: ARTIFICIAL INTELLIGENCE II
Session Chair: Dr. Abe Waksman

Learning procedures from interactive natural language instructions 207

Fuzzy Logic, Neural Networks, and Soft Computing 217

Hybrid Knowledge Systems 218

Two Frameworks for Integrating Knowledge in Induction 226

Discovering operating modes in telemetry data from the Shuttle Reaction Control System 234

Self-Calibrating Models for Dynamic Monitoring and Diagnosis 245

Session A3: ARTIFICIAL INTELLIGENCE III
Session Chair: Dr. Peter Friedland

Filtering as a Reasoning-Control Strategy: An Experimental Assessment 253

Translation Between Representation Languages 261

A Machine-Learning Apprentice for the Completion of Repetitive Forms 268

Automated Knowledge-Base Refinement 276

Improving the Explanation Capabilities of Advisory Systems 284

Recursive Heuristic Classification 301

Session A4: ARTIFICIAL INTELLIGENCE IV
Session Chair: Dr. Abe Waksman

Agent Oriented Programming 303

PI in the Sky: The Astronaut Science Advisor on SLS-2 304

Engineering Design Knowledge Recycling in Near-Real-Time 313

A Toolbox and a Record for Scientific Model Development 321

SECTION III: HUMAN FACTORS

Session H1: GROUND OPERATIONS TEAMS
Session Chair: Dr. Kristin Bruno

HFE Safety Reviews of Advanced Nuclear Power Plant Control Rooms 329

Using Task Analysis to Understand the Data System Operations Team 337

The Virtual Mission Operations Center 345

Mission Operations and Command Assurance: Flight Operations Quality Improvements 356

Session H2: ENHANCED ENVIRONMENTS
Session Chair: Maj. Gerald Gleason

Call sign intelligibility improvement using a spatial auditory display 363

Laboratory and In-Flight Experiments to Evaluate 3-D Audio Display Technology 371

Fusion Interfaces for Tactile Environments: An Application of Virtual Reality Technology 378

Virtual Reality as a Human Factors Design Analysis Tool: Macro-Ergonomic Application Validation and Assessment of the Space Station Freedom Payload Control Area 388

A Rapid Algorithm for Realistic Human Reaching and Its Use in a Virtual Reality System 389

Session H3: PSYCHOPHYSIOLOGY, PERFORMANCE, AND TRAINING TOOLS
Session Chair: Dr. James Whitely

Autogenic-Feedback Training Improves Pilot Performance During Emergency Flying Conditions 395

Implementing Bright Light Treatment for MSFC Payload Operations Shiftworkers 404

Flight Controller Alertness and Performance During MOD Shiftwork Operations 405

Automating the Training Development Process for Mission Flight Operations 417

Session H4: MODELING IN SUPPORT OF OPERATIONS AND ANTHROPOMETRY
Session Chair: Ms. Barbara Woolford

Application of Statistical Process Control and Process Capability Analysis Procedures in Orbiter Processing Activities at the Kennedy Space Center 427

Task Network Models in the Prediction of Workload Imposed by Extravehicular Activities During the Hubble Space Telescope Servicing Mission 432

A Human Factors Evaluation Using Tools for Automated Knowledge Engineering 437

An Overview of Space Shuttle Anthropometry and Biomechanics Research with Emphasis on STS/Mir Recumbent Seat System Design 442

Anthropometric Accommodation in USAF Cockpits 447

Session H5: **BEING THERE: PROTOTYPE AND SIMULATION FOR DESIGN**
Session Chair: **Dr. Jane Malin**

End Effector Monitoring System: An Illustrated Case of Operational Prototyping 461

The PLAID Graphics Analysis Impact on the Space Program 468

Human Factors Involvement in Bringing the Power of AI to a Heterogeneous User Population 475

A Comparison of Paper and Computer Procedures in a Shuttle Flight Environment 483

SECTION IV: **LIFE SUPPORT**

Session L1: **SPACE PHYSIOLOGY**
Session Chair: **Ms. Susan Fortney**

Gravity, The Third Dimension of Life Support in Space 493

The U.S. Navy/Canadian DCIEM Research Initiative on Pressure Breathing Physiology 494

Response to Graded Lower Body Negative Pressure (LBNP) After Space Flight 495

Dual-Cycle Ergometry as an Exercise Modality During Prebreathe with 100% Oxygen 496

Exercise with Prebreathe Appears to Increase Protection from Decompression Sickness 497

Orthostatic Responses to Dietary Sodium Restriction During Heat Acclimation 501

Session L2: **MEDICAL OPERATIONS**
Session Chair: **Lt. Col. Roger Bisson**

Telemedicine, Virtual Reality, and Surgery 509

Total Hydrocarbon Analysis by Ion Mobility Spectrometry 522

The Effectiveness of Ground Level Post-Flight 100% Oxygen Breathing as Therapy for Pain-Only Altitude Decompression Sickness (DCS) 532

Emergency Medical Services 538

Preliminary Health Survey Results from Over 250 High Flyer Pilots with Occupational Exposures to Altitudes Over 60,000 Feet 540

Session L3: TELEMEDICINE
Session Chair: Dr. Gerald Taylor

Quantitative 3-D Imaging Topogrammetry for Telemedicine Applications 553

IMIS – An Intelligence Microscope Imaging System 554

The Portable Dynamic Fundus Instrument: Uses in Telemedicine and Research 555

Technical Parameters for Specifying Imagery Requirements 557

Session L4: THERMAL STRESS
Session Chair: Col. Gerald Krueger

Personal Cooling Systems: Possibilities and Limitations 559

Modeling HEAT Exchange Characteristics of Long Term Space Operations: Role of Skin Wettedness and Exercise 560

Hydration and Blood Volume Effects on Human Thermoregulation in the Heat: Space Applications 567

Prediction Modeling of Physiological Responses and Human Performance in the Heat with Application to Space Operations 574

Session L5: BIOMEDICAL RESEARCH AND DEVELOPMENT
Session Chair: Capt. Terrell Scoggins

Evaluation of a Liquid Cooling Garment as a Component of the Launch and Entry Suit (LES) 583

La Chalupa-30: Lessons Learned from a 30-day Subsea Mission Analogue 584

Comparative Performance of a Modified Space Shuttle Reentry Anti-G Suit (REAGS) With and Without Pressure Socks 596

Current and Future Issues in USAF Full Pressure Suit Research and Development 597

Advanced Integrated Life Support System Update 598

Session L6: TOXICOLOGY AND MICROBIOLOGY
Session Chair: Mr. Richard Sauer

Continuous Monitoring of Bacterial Attachment 599

Characterization of Spacecraft Humidity Condensate 600

Computation of Iodine Species Concentrations in Water 601

A Volatile Organic Analyzer for Space Station: Description and Evaluation of a Gas Chromatography/Ion Mobility Spectrometer 602

Safety Concerns for First Entry Operations of Orbiting Spacecraft 603

SECTION V: SPACE MAINTENANCE AND SERVICING

Session S1: SPACE STATION MAINTENANCE
Session Chair: Mr. Kevin Watson

Unique Methods for On-Orbit Structural Repair, Maintenance, and Assembly 605

On-Orbit NDE—A Novel Approach to Tube Weld Inspection 611

Force Override Rate Control for Robotic Manipulators 619

NASA Shuttle Logistics Depot Support to Space Station Logistics 628

Session S2: SPACE MAINTENANCE AND SERVICING
Session Chair: Mr. Charles Woolley

Graphical Programming: A Systems Approach for Telerobotic Serving of Space Assets 629

Diagnosing Anomalies of Spacecraft for Space Maintenance and Servicing 641

The SCRAM Tool-Kit 652

SpaceHab I Maintenance Experiment 661

A Standard Set of Interfaces for Serviceable Spacecraft by the NASA Space Assembly and Servicing Working Group 666

INTRODUCTION

Kumar Krishen, Ph.D.

The Space Technology Interdependency Group (STIG) was established in May 1982 to identify and promote the pursuit of new opportunities for cooperative relationships and monitor ongoing cooperative activities between the National Aeronautics and Space Administration (NASA) and the U.S. Air Force. In the past 5 years, the U.S. Army, U.S. Navy, Advanced Research Projects Agency (ARPA), Ballistic Missiles Defense Organization (BMDO), and Department of Energy (DOE) have joined the STIG Steering Committee. The goals of STIG are to provide advocacy, oversight, and guidance to facilitate and encourage technology applications. The members of the Steering Committee are from NASA Headquarters' executive staff to provide technical and management expertise to evaluate programs and to suggest new approaches to foster interdependencies.

Eight technical committees have been established by the STIG Steering Committee (fig. 1). Members of these committees are selected from participating field organizations. The cochairpersons for the technical committees are nominated by Steering Committee members and are approved by Steering Committee cochairpersons. The U.S. Air Force Materiel Command Deputy Chief of Staff for Technology (HQ AFMC/XT) and the NASA Associate Administrator for Advanced Concepts and Technology (HQ NASA OACT/Code C) serve as cochairpersons for the STIG Steering Committee. For the purposes of planning and execution, interdependent programs are defined as having some degrees of overlap in stated agency program and/or technical goals, as outlined in a jointly developed program plan. In executing interdependent programs, complementary synergistic results benefit all participating agencies.

The STIG Operations Committee (SOC) goals have been developed through several interactions in the past 5 years. Current goals of this committee include: (1) to identify and characterize interdependent activities; (2) to encourage interdependent programs; (3) to interchange technical and programmatic information and share lessons learned; (4) to identify critical voids and nonproductive overlaps in technology programs; (5) to develop technology area road maps which identify interrelationship of activities and sharing of resources between participating organizations; and (6) to promote technology transfer to industry and academic institutions. The implementation strategy for SOC is as follows:

- Conduct STIG Operations, Applications and Research (SOAR) Symposium and Exhibition on a yearly basis
  - Include technical review of interdependent programs
  - Identify future interdependent programs
  - Identify areas of concern
  - Include industry and academia

- Organize five subcommittees under SOC
  - Robotics and Telepresence
  - Automation and Intelligent Systems
  - Human Factors
  - Life Support
  - Space Maintenance and Servicing (effective 9/93, this subcommittee is being replaced with a new subcommittee named Guidance, Navigation and Control which will include on-orbit operations only)

- Conduct two SOC meetings on a yearly basis
  - Review operations R&T plans, resources, progress with NASA, DOD, and DOE
  - Develop and maintain list of descriptions of interdependent programs
  - Encourage and recommend interdependent programs

- Facilitate communications of R&T results in operations area across agencies and various centers within these agencies involved in this R&T and also to industry and academic institutions

- Include both ground and space operations R&T in SOC activities

- Provide interface with NASA, DOD, and DOE Operations Technology Thrusts and other STIG committees, specifically Information Collection, Processing and Transfer Committee

The overall organization and membership of the SOC are shown in figures 2 through 7.

The Seventh Annual SOAR Symposium and Exhibition was held on August 3-5, 1993, at the NASA Johnson Space Center. The symposium contained 25 technical sessions in 5 discipline areas: Robotics and Telepresence, Automation and Intelligent Systems, Human Factors, Life Support, and Space Maintenance and Servicing. Approximately 121 technical papers and presentations were included in the program. A Plenary Session on *Operations Experiences* and a panel discussion on *Operations Challenges* highlighted the identification of a road map for future technology thrusts. As a part of these discussions, a STIG operations research and technology process chart was presented by this author to highlight concerns for streamlining a process approach to operations technology development and deployment (fig. 8). Seventeen exhibitors supported SOAR '93. We had over 300 registered SOAR '93 participants, with an additional 200 participants for exhibition viewing. Drs. Aaron Cohen, Earl Good, and Melvin Montemerlo provided keynote speeches to paint the national picture for space programs and needed technology developments. This proceeding captures most of the presentations.

The SOAR '93 program has received extremely favorable comments from the participants and exhibitors. Credit for the achievements belongs to the program committees, listed in figure 9. Your comments and suggests to improve the SOC or SOAR programs are always welcome and should be addressed to:

Dr. Kumar Krishen
Co-chair, SOC
Code IA4
NASA Johnson Space Center
Houston, TX 77058.

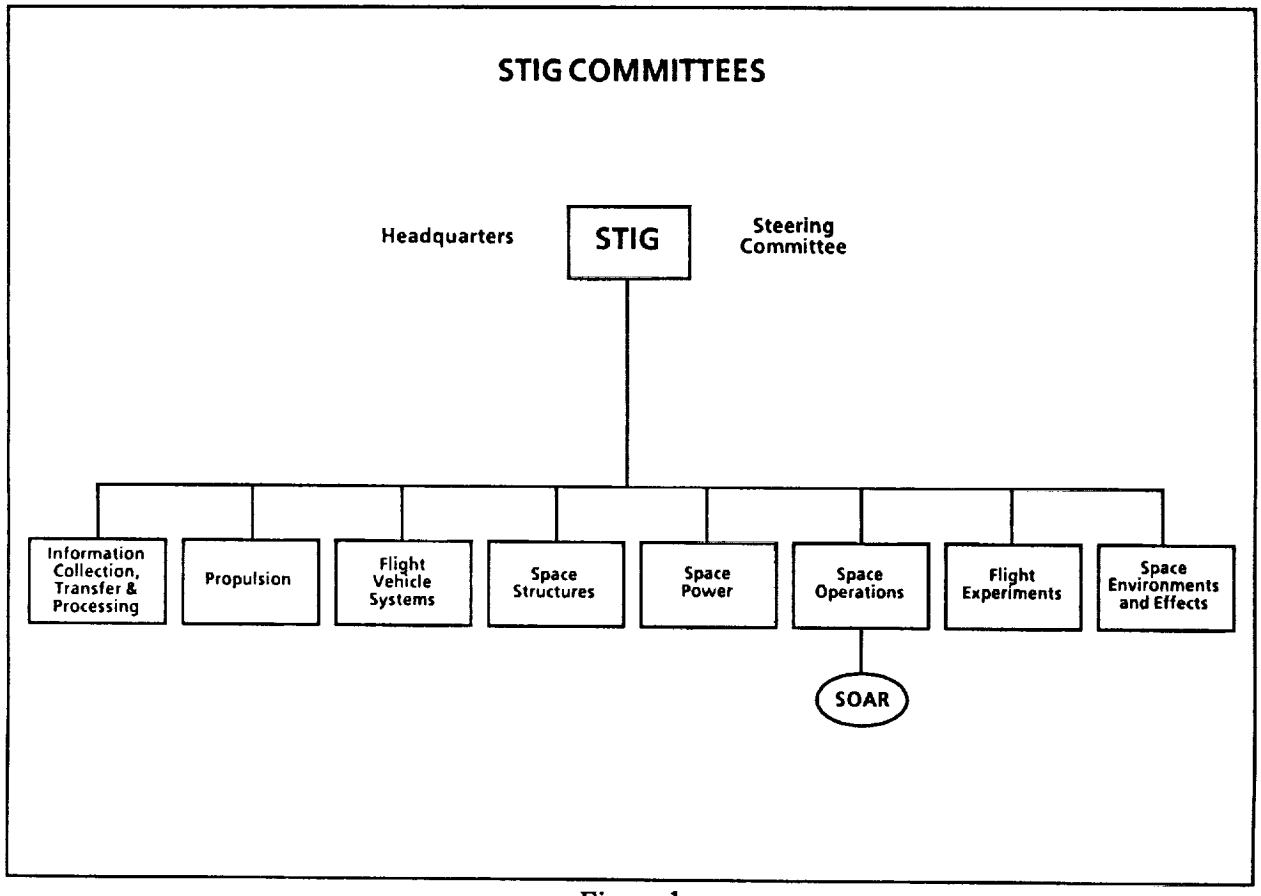


Figure 1.

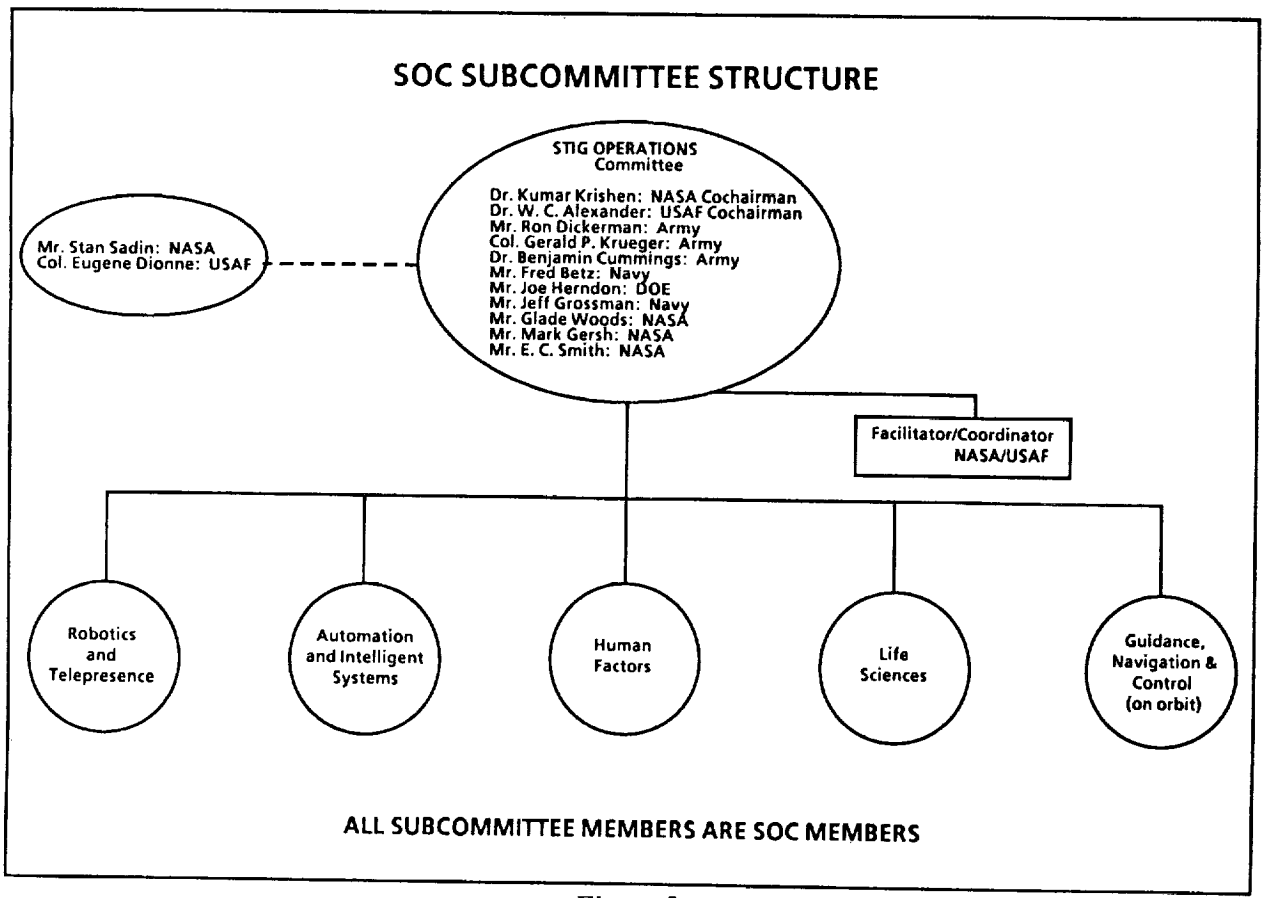


Figure 2.

- Robotics and Telepresence Subcommittee
  - Scope
    - Telepresence, teleoperation, telerobotics, autonomous robotics
    - Space maintenance and assembly, planetary exploration, terrestrial applications
    - Dexterous manipulation, navigation, perception, and control
  - Membership
    - Capt. Paul Whalen*/AF Armstrong Lab
    - Dr. Charles Weisbin*/NASA JPL
    - Mr. Ed Alexander/AF CESA
    - Mr. William Helms/NASA KSC
    - Mr. Joe Herndon/DOE ORNL
    - Ms. Elaine Hinman-Sweeney/NASA MSFC
    - Mr. Mark Jaster/NASA GSFC
    - Capt. Ron Julian/AF Armstrong Lab
    - Mr. David Lavery/NASA HQ
    - Dr. Michael McGreevy/NASA ARC
    - Dr. Teresa McMullen/ONR
    - Mr. Jack Pennington/NASA LaRC
    - Mr. Charles Price/NASA JSC
    - Mr. Eric Rhodes/NASA KSC
    - Mr. Wayne Schober/NASA JPL
    - Mr. Charles Shoemaker/ARL
    - Capt. Gary E. Yale/AF Phillips Lab

* Co-chairpersons

Figure 3.

- Automation and Intelligent Systems Subcommittee
  - Scope
    - Knowledge-based systems/expert systems
    - Artificial intelligence
    - Neural networks
    - Fuzzy logic
    - Vehicle health monitoring
  - Membership
    - Capt. Jim Skinner*/AF Wright Lab
    - Dr. Peter Friedland*/NASA ARC
    - Capt. Mary Boom/AF Phillips Lab
    - Dr. Richard Doyle/NASA JPL
    - Mr. William Helms/NASA KSC
    - Ms. Kathleen Jurica/NASA JSC
    - Mr. Ralph Kissel/NASA MSFC
    - Dr. Melvin Montemerlo/NASA HQ
    - Mr. James Overholt/TACOM
    - Mr. Robert Savely/NASA JSC
    - Ms. Nancy Sliwa/NASA KSC
    - Dr. Abraham Waksman/AFOSR

* Cochairpersons

Figure 4.

- Human Factors Subcommittee
  - Scope
    - Human performance measurement, modeling, and prediction
    - Extra- and intra-vehicle operations
    - Human-machine interactions
    - Training systems
    - Workload and scheduling
    - Virtual environments/virtual reality
    - Crew selection, composition, and coordination
  - Membership
    - Col. Gerald P. Krueger*/USA RIEM
    - Dr. Mary Connors*/NASA ARC
    - Dr. Kristin Bruno/NASA JPL
    - Dr. Carl Englund/NRaD
    - Lt. Col. Gerald Gleason/AF Armstrong Lab
    - Dr. Jonathon Gluckman/Navy Air Warfare Center
    - Mr. Joseph Hale/NASA MSFC
    - Dr. Jane Malin/NASA JSC
    - Dr. Richard Monty/ARL/HRED
    - Dr. Sylvia Sheppard/NASA GSFC
    - Dr. James Walrath/ARL/HRED
    - Mr. William B. Williams/NASA KSC
    - Ms. Barbara Woolford/NASA JSC

* Co-chairpersons

Figure 5.

- Life Sciences Subcommittee
  - Scope
    - Life support
    - Health systems
    - Biomedical research
    - Medical operations
    - Space radiation effects
  - Membership
    - Dr. Andrew Pilmanis*/AF Armstrong Lab
    - Dr. Gerald Taylor*/NASA JSC
    - Lt. Col. Roger U. Bisson/AF Armstrong Lab
    - Dr. Malcolm M. Cohen/NASA ARC
    - Dr. Jerry Homick/NASA JSC
    - Col. Gerald P. Krueger/USA RIEM
    - Dr. Gregory Nelson/NASA JPL
    - Capt. Terrell Scoggins/AF Armstrong Lab
    - Dr. C. Lewis Snead/DOE BNL
    - Dr. Phil Whitley/Navy Air Warfare Center

* Co-chairpersons

Figure 6.

- Space Maintenance and Servicing
  - Scope
    - Maintenance and repair operations
    - Assembly operations
    - Servicing operations
    - Fault detection
    - Nondestructive evaluation
  - Membership
    - Vacant*/DOD
    - Mr. Chuck Woolley*/NASA JSC
    - Mr. Jerry Borrer/NASA JSC
    - Mr. Tom Bryan/NASA MSFC
    - Mr. John Cox/USAF SSD
    - Mr. Bill Eggleston/NASA JSC
    - Mr. Jeffrey Hein/NASA JSC
    - Dr. Neville Marzwell/NASA JPL
    - Mr. Don Nelson/NASA JSC

NOTE: Effective 9/93, this subcommittee is being replaced by another subcommittee named Guidance, Navigation & Control and will include on-orbit operations only.

Co-chairpersons and subcommittee members are in the process of being formed.

* Co-chairpersons

Figure 7.

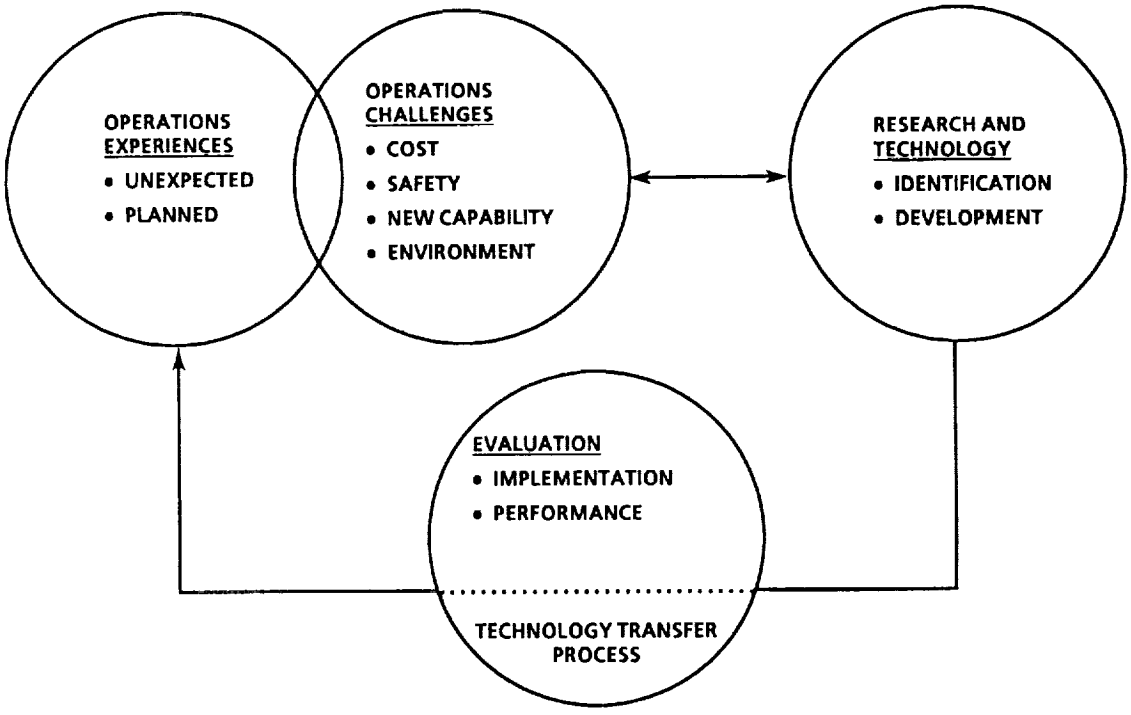


Figure 8.

SOAR '93 will include USAF and NASA programmatic overviews, panel sessions, exhibits, and technical papers in the following areas:

- Robotics and Telepresence
- Automation and Intelligent Systems
- Space Maintenance and Servicing
- Human Factors
- Life Support

Exhibit Hours

Tuesday, August 3	10:00 am – 7:00 pm
Wednesday, August 4	8:00 am – 7:00 pm
Thursday, August 5	8:00 am – Noon

Welcome/Opening Addresses (August 3, 8:30 am – 9:00 am)

Mr. Aaron Cohen, NASA JSC
Dr. W. C. Alexander, AL/XP
Dr. Kumar Krishen, NASA JSC

Plenary Session (August 3, 9:00 am)
Operations Experiences

Mr. John Muratore, NASA JSC
Lt. Col. Roger Bisson, USAF
Dr. Howard Schneider, NASA JSC
Maj. Mark Pestana, USAF
Mr. Richard Hieb, NASA Astronaut

Panel Discussion (August 3, 3:30 pm – 5:00 pm)
Operations Challenges

Moderator:	Dr. Kumar Krishen
Panelists:	Dr. Melvin Montemerlo, NASA HQ Gael Squibb, NASA JPL Mr. John Muratore, NASA JSC Maj. Kory Cornum, USAF

Keynote Dinner Session (August 4, 6:00 pm – 9:00 pm)

Welcome and Opening Remarks	Mr. Aaron Cohen Director NASA JSC
Keynote Speakers:	Mr. Gregory Reck, NASA HQ Dr. R. Earl Good, USAF

Symposium Coordinators

Symposium Cochairs:	• Dr. Kumar Krishen NASA/JSC
	• Dr. W. C. Alexander USAF AL/XP
Technical Coordinators:	• Mr. Robert Savely NASA/JSC
	• 2LT Catherine Moore AL/XPT
Administrative Cochairs:	• Ms. Carla Armstrong I-NET, Inc.
	• Ms. Lana Arnold Lockheed/ESC
	• Mr. Dick Rogers Lockheed/ESC
	• Ms. Stancie Chamberlain University of Houston–Clear Lake
Exhibit Cochairs:	• Mr. Chris Ortiz NASA/JSC
	• Mr. Ellis Henry I-NET, Inc.
	• Ms. Resa Ott University of Houston–Clear Lake

Technical Area Coordinators

	USAF	NASA
Robotics and Telepresence	Capt. Ron Julian AL/CFBA (513) 255-3671	Dr. Charles Weisbin NASA JPL (818) 354-2013
Automation and Intelligent Systems	Capt. Jim Skinner WL/AAA-1 (513) 255-5800	Dr. Peter Friedland NASA ARC (415) 604-4277
Human Factors	Col. Donald Spoon AL/CF (513) 255-5227	Dr. Mary Connors NASA ARC (415) 604-6114
Life Support	Dr. Andrew Pilmanis AL/CFTS (512) 536-3545	Dr. Gerald Taylor NASA JSC (713) 244-8796
Space Maintenance and Servicing		Mr. Charles Woolley NASA JSC (713) 244-8354

Figure 9.

WELCOME / OPENING ADDRESSES

WELCOME / OPENING ADDRESSES

Dr. W. C. Alexander
Armstrong Laboratory

Welcome each and every one of you to the Seventh Annual Space Operations, Applications and Research Symposium (SOAR '93). It's good to be back. We had a great meeting last year. We had a good session across the year with our Space Operations Committee. We're really pleased that we were able to take the suggestions of our committee to rearrange this first day of the conference symposium and look more in depth at the thing that we really do for a living, which is operations. Each and every one of us has some tie into the operational community across the DOD and, certainly, with our colleagues here at NASA.

The SOAR conference each year is sponsored jointly by NASA and by the U.S. Air Force. NASA's turn was 1993. This conference is because of Dr. Kumar Krishen and his staff. A good conference; a good schedule; a good agenda laid out; and I think a good time for everyone here. I want to keep my remarks brief because we have a tremendous program in store for the day. At this time I'd like to welcome and introduce to you Dr. Kumar Krishen, the NASA host and cochair for this year's conference.

WELCOME / OPENING ADDRESSES

Dr. Kumar Krishen
NASA Johnson Space Center

This morning I have the pleasant duty of introducing to you our keynote speaker. Our opening speaker this morning exemplifies the excellence achieved when the experiences of Government, industry, and academia are combined – and, in this case, in one person: Aaron Cohen. After more than 30 years with NASA, and prior to that with industry, the Director of the Johnson Space Center – Mr. Aaron Cohen – is retiring this month to become the Zachry Professor of Engineering at his alma mater, Texas A&M University, where he will be developing new educational initiatives in the area of systems engineering. Because he is considered such a valuable resource, he will also serve as special consultant to the NASA Administrator on human space flight as well as on research and technology.

Mr. Cohen has had a major impact on the future of human space flight since he came to what was then the Manned Spacecraft Center in 1962. He has served in key leadership roles, where his efforts were critical to the success of all six lunar landings. He managed the hardware and software designed to provide guidance, navigation, and control for the command and service module and for the lunar module. He also served as manager for the command and service modules for the Apollo spacecraft program. From 1972 to 1982, he served as manager of the Space Shuttle Orbiter Project and directed the design, development, production, and testing of the Orbiter. After completion of the Orbiter test flight, Mr. Cohen became Director of Research and Engineering and was responsible for all engineering and space and life sciences research in support of major human flight programs at JSC. In 1986, Mr. Cohen was appointed JSC Director and helped to guide the Space Shuttle Program back to flight after the STS 51-L accident. He also served as Acting Deputy Administrator of NASA from March 1992 to March 1993.

Dr. Aaron Cohen
Director, NASA Johnson Space Center

THE FINE LINE
Opportunities and Obstacles in Space Operations

Good morning. Welcome to the Seventh Annual SOAR Symposium; and, to our guests, welcome to the Johnson Space Center. During the next few days, your sessions, your speakers, and, most importantly, your personal discussions with each other will provide a wealth of technical knowledge about improving operations in space and on the ground. I would like to thank the symposium chairmen, Dr. Kumar Krishen of JSC and Dr. Carter Alexander of the U.S. Air Force, and all the administrative and exhibit teams for working so hard to bring about this highly respected conference.

I wear two hats while I attend your sessions this week – one as a NASA manager and another as an eager new academic. As you have heard, I will leave JSC in a few weeks to become the Zachry Professor of Engineering at Texas A&M University. My 31 years here at JSC have been extremely meaningful and fulfilling. Now, I have the opportunity to work with the next generation of space leaders and pioneers. I want to share with them the excitement, the experiences, and some lessons learned so these up-and-coming professionals will be ready to develop the most efficient, cost-effective, and successful space flight programs possible. That is why I believe this SOAR '93 symposium is so important.

SOAR provides the forum where Government agencies, industry, academia, and other researchers can come together to share technical information and lessons learned, to discover areas where cooperative efforts can enhance this Nation's space goals, and to identify critical voids or unproductive duplications that could limit our operational effectiveness. This need for cooperation is even more relevant now that the Cold War has ended. As the walls to international cooperation and communication have fallen, we in the Government also must chip away at existing barriers that previously limited joint projects and information exchange.

Each of our agencies sits under a fiscal microscope. We constantly justify our programs to our Washington leaders, who demand an immediate return on taxpayers' dollars. To that end and for our future, we must identify technologies up front that benefit the private and public sectors, and work to jointly develop these. We must look at impediments in our procurement systems and find better ways to work with industry to bring greater cost savings and process efficiencies to our programs.

The interagency and extra-agency connections you make during this conference are keys to solving the many challenges facing our Nation's aerospace future. Operations is the thin line in human space flight, the thin line between mission success and mishap, crew productivity and crew survival. New demands of human space flight require we reduce costs while increasing spacecraft efficiencies and safety – and that's no easy task.

Over the next few years, JSC increasingly will lead operations for simultaneous Shuttle and Space Station missions. More and more, we will deal with longer duration experience in space as well as with multinational crews. Both necessitate paradigm shifts in the way we plan and conduct

missions. Crews will increasingly perform more complex on-orbit maintenance and servicing – as with the upcoming Hubble Space Telescope repair mission and future Space Station upkeep – which require substantial training and operational advances.

The Johnson Space Center remains on the cutting edge of operations technology for human space flight. JSC expertise covers many areas essential to the future of operations: robotics and telepresence, automation and intelligent systems, human factors, life sciences, and space maintenance and servicing. In order to successfully reach these goals, a prime concern of the JSC team is to keep the Shuttle flying safely and effectively. While the largest portion of our current Center budget is devoted to Shuttle, this investment ultimately allows for many productivity enhancements.

Today, JSC supports about eight Shuttle flights each year with relatively the same number of people as during the Apollo Program, when we flew two missions per year. That efficiency is due to many operational improvements, particularly in the area of information management. Through ongoing mission operation efficiency planning, a 30% mission operations cost reduction already has been achieved with an additional 5% targeted by 1999. Our engineers and scientists are looking at innovations to mission training and operations that also have exciting applications in the commercial world.

The evolving field of Intelligent Computer Aided Training – or ICAT – captures the expertise of a human teacher in a computer program. The computer understands training protocols, keeps track of trainee progress, and critiques efforts. We have tested the system for the first time with the recent STS-57 Spacehab mission. While some crews prefer to work with real hardware in simulations, other crews want to have ICAT trainers on their desks to supplement practice as their schedules permit.

The futuristic human-machine interface of Virtual Reality potentially can expand training opportunities for crews, flight controllers, and other workers inside and outside the aerospace field. Right now, JSC engineers are assessing a V-R trainer that allows the flight controller to get the "real feel" of doing a space walk. We cannot afford the time or cost to train flight controllers on actual flight hardware. But with a V-R system, they can "experience" the difficulties astronauts face during a complicated EVA, such as the Hubble telescope repair, and, therefore, develop more realistic procedures and contingencies for such missions.

We already are using expert systems in the Mission Control rooms to monitor Shuttle systems, which offer a great improvement over Apollo and even early Shuttle operations hardware. But if you go into the flight control or support rooms, you will see one thing has not changed since Gemini days – the reams of paper required by each controller to keep track of the mission.

We at JSC are testing a hyper-manual system that could revolutionize mission operations. Rather than supplying flight controllers with bookcases of 3-ring binders for a mission, one electronic version in portable workstations could be used by all, individualized to each controller's specific needs. The electronic manual works like a user-friendly paper version. Controllers can scribble notes on the side, highlight important information, electronically "paper clip" pages for easy reference, and even "rip out" unneeded sections. The electronic version will let the reader flip to a different manual with a click of a key, and will provide simultaneous real-time updates.

These are technologies that can increase productivity and training opportunities in the private and public sectors – the return on investment I mentioned that is so important in these tough fiscal times. For example, the JSC ICAT system is being adapted as an Intelligent Physics Tutor for high school students and as an Adult Literacy Tutor to train the functionally illiterate in prisons and

through nonprofit organizations. Both systems are in the prototype stage and are on their way toward commercialization.

JSC also is the NASA focal point for human life science research and technology, an essential operations area. We must better understand how humans adapt to living and working in space and improve the means to ensure their health, safety, comfort, and productivity. Without such research, no extended space exploration by humans will be possible.

A recent cooperative agreement between NASA and the Texas Medical Center exemplifies the type of cooperative efforts that SOAR advocates. The Texas Medical Center is internationally respected for treating sick bodies in a normal terrestrial environment, while NASA works with very healthy bodies in the abnormal environment of space. When we join forces to understand how the human body works and adapts both in space and on the ground, we have a formidable team. Imagine the medical breakthroughs that await from such an alliance. We already have teamed with famed heart surgeon Dr. Michael DeBakey to apply JSC spacecraft technology to the development of a new potentially lifesaving heart pump.

A fine line exists between opportunity and obstacles in our Nation's space future. Our success stems not so much from how well we deal with current space operations issues, but rather from how well we identify tomorrow's operations concerns and take the foresighted actions to solve them today. SOAR '93, the STIG Operations Committee, and our Government-industry-and-academia teams are in the forefront of meeting that challenge now.

Thank you and best wishes for an insightful meeting.

PLENARY SESSION HIGHLIGHTS

OPERATIONS EXPERIENCE

Lt. Col. Roger Bisson, USAF
Dr. Howard Schneider, NASA JSC
Mr. Richard Hieb, NASA Astronaut
Mr. John Muratore, NASA JSC
Maj. Mark Pestana, USAF

ALEXANDER: The first speaker in operations experiences comes from Armstrong Laboratory – formerly the School of Aerospace Medicine over at Brooks Air Force Base. Lt. Col. Roger Bisson is a B-52G pilot who flew almost 1000 hours in aircraft, went to medical school and became a physician, stayed in the U.S. Air Force, and has become a Flight Surgeon with a Board Certification Residency behind him in aerospace medicine. Roger epitomizes operations and operations experiences. He recently published results of an operations study in heavy airlift to the Desert Storm activity. He flew with the C-5 and C-141 crews in airlift throughout that entire campaign. He now will go from multicrew studies to a single-seat fighter study, assisted by Maj. Kory Cornum. Roger's a private pilot. He's been in the strategic reconnaissance business out at Beale and he brings to us a wealth of this experience.

BISSON: It seems as if I've been away from operations for some period of time, but one of the nice things about Armstrong Laboratory is that we maintain an operations perspective and can transition some of our work very quickly into the operational field.

I'm going to talk about the use of digital flight data to not only look at aircraft performance, but how the human impacts on that performance and what digital flight data means in terms of human factors, and how to transition that technology as a really strong defense conversion opportunity that will be useful not only to the U.S. Air Force but certainly to industry and NASA as well.

The purpose behind this presentation is to talk a little bit about Global Reach-Global Power, which is the U.S. Air Force's vision right now – the operational vision – of what the U.S. Air Force is going to be about for at least the next 10 years or on into the future. That vision particularly impacts our Sustained Operations Branch at Armstrong Laboratory, because Global Reach speaks very strongly about what our capacity was during Desert Storm as far as conducting a war so far from our own homeland. Global Reach in these terms talks about

BISSON: (Cont'd)

taking conventional operations – long-range bombing, B-1, B-2, B-52 type missions – and holding the same targets at risk that we held in the first days of the air war for a future requirement. We don't anticipate that our next enemy will give us 6 months in the desert to wind up and give a punch. And so, present plans have to call for being able to hold that same sort of target set at risk from CONUS [Continental U.S.]. This is something that had been contemplated before, but not in terms of a sustained conventional bombing campaign from our own shores. We've never had to face that kind of a thing in the United States, but the world is getting smaller.

We started out in the C-141 during Desert Shield trying to look at how fatigue impacted long-duration flights, and from there comes the story of digital flight data. As we got back to peacetime operations, we started looking at the Global Power side of the equation. That harkens back to my own B-52 days in a large study that we just completed in the B-1B, basically proving the operational concept of sustained conventional bombing from the U.S. It may not be as far-reaching a thought as you may think as we consider actions in other parts of the world that may not give us overflight rights or landing rights for some of the places we would like to conduct our operations.

For our Sustained Operations Branch in talking about operational long-duration missions, this is the enemy really: it's fatigue. It goes by many aliases, among them exhaustion, weariness, lassitude, apathy, all those other things. Once again, we try to think in terms of fatigue in an operational sense and that there's an intelligence threat in that it causes errors in judgment and accuracy, response times, blurred vision, muscular weakness, and all these other types of things. In our theater of operations, we include circadian loss and circadian dysrhythmia, sleep loss. Our objectives certainly are to contain the threat and to neutralize the enemy. Our fatigue countermeasures basically focus on two aspects, in that we can either promote vigilance or promote crew rest to enhance performance for the sustained operations regime.

Our deliverable weapons in this area to promote vigilance and crew rest include work in intelligent tutoring systems as well as in team and group operations and looking at performance in those operations. We have some exciting work going on right now in Bright Lights and Tyrosine and Modafinil, combining all three of those agents in order to rapidly shift circadian rhythms. We use exercise/ nutrition. But, what I'm going to gradually focus back on is using the digital flight data as a means of enhancing aircrew performance and training and safety, and how that represents a defense conversion opportunity.

BISSON: (Cont'd)

For promoting crew rest drugs aren't that popular these days, but we still continue to do work with Restoril and Melatonin. It's very important to take our research and transition it to the operational world as quickly as possible.

Our lead times in many of our technologies right now are on the order of 6 months. The B-1 study that we just completed at Ellsworth and Dyess Air Force Bases in their simulators was designed and conducted and implemented, and it started impacting their operations in less than a 6-month time frame. It involved several hundred hours of simulation time, writing the reports, and getting our findings actually into operational plans.

Current operations within the Sustained Operations Branch have to do with Melatonin, quality of sleep, shift work and fatigue. We also just completed another study with air traffic controllers, looking at how their shift work schedules affect their efficiency. We're about to embark on a fighter pilot fatigue study that involves looking at combat air patrol and deployment schedules. I don't think anyone, 6 months before Desert Shield kicked off, would have said that we'd be triple turning some of the fighters and having fighter missions that are as long as 6 and 8 hours back to back for some of the crew members having flying duty days that, in some cases, approached or exceeded 16 hours in single-seat aircraft. The C-141 fatigue study has much to do with our vision for the future.

Warbreaker is an exercise where teams train with linking networks up across the U.S. for simulating wars but actually have pilots in the skies with aggressors. It's just totally integrating a battle using resources across the country for training purposes as well as for planning. We're doing a lot of work in computer modeling of fatigue and performance. We're in the process of coming up with a fatigue management doctrine that we think will be useful for schedulers and mission planners in how to impact fatigue.

The B-1B mishap at Ellsworth Air Force Base in South Dakota was clearly a human factors mishap. It illustrated what happens when you're in the weather and manage somehow to get below your minimum descent altitude and find out that there are telephone poles growing up a little bit higher than what your altitude is. The digital flight data from the crash was critical in reconstructing the mishap scenario. That impacted me in looking at it. I wasn't looking at what the control surfaces were doing as much as what the humans inside that aircraft were doing in order to survive that contingency.

About the time that Desert Shield kicked off I was transitioning to work at Armstrong Laboratory. The C-141 fatigue study had actually already been

BISSON: (Cont'd)

designed by the time I came on board. I talked to Dr. Storm and suggested that the capability of taking the digital flight data off of the C-141 existed and we could probably do it before the airplanes crashed. These data would reflect, not so much just what the aircraft was doing, but whether the fatigue of the aircrew was starting to affect their performance. Lots of quick legwork, and we managed to get the boxes that we needed and the technology in place for a study to actually download the digital data at the end of every flight.

Historical uses for digital flight data, of course, involve aircraft accident investigation. Over the years the number of parameters that are being monitored have increased, and the FAA and other regulatory agencies have increased the requirements for the ability to capture these data.

The opportunity here for the U.S. Air Force is a defense conversion opportunity. I think the implications and the opportunities for using digital flight data for safety, for training, and for monitoring trends are there also.

The descent approach to landing phase of a flight is perhaps the most dangerous because it is at the end of the mission with the C-141 crews who have been flying up to 120 hours in the last 30 days. On the digital data tape, we can get such things as approach speed, their rate of descent, their bank angles, and where they are on their glide path or on their localizer.

The digital flight data recorder system in the C-141 is not as sophisticated as some. It took aircraft inputs from airspeed, heading, pitch, roll, flap, and spoilers and about 20 some odd parameters and put them into a flight data acquisition unit. The digital flight data recorder could easily be connected up to a copy recorder and, at the end of each flight, you could capture 30 hours of flight data.

The box that we needed to use to capture that data is not much bigger than a little suitcase, and it's getting smaller. So the technology to capture these data is something that could easily be built into aircraft. There are about 25 non-U.S. carriers who are capturing digital flight data for some of the purposes that I'm talking about. However, the U.S. industry still is in its infancy as far as being willing to adopt this technology.

When you capture the data, basically you end up with spreadsheets. As you start to look at those numbers, they really start to tell a story. Now, I take that and say it's a human action that you can interpret about what airspeed they lowered the gear and factors like that. But, engineers can take a look at it and maybe design it so that that pitchup that you see from each end of the gear is not part of

BISSON: (Cont'd)

the aircraft characteristics. And, maybe the pilots need to be aware that, in fact, this pitchup occurs so that they can counteract it or some automated systems could be built in to automatically counteract it for them.

With the onset of artificial intelligence, the potential for a third pilot basically being in the back is there. The U.S. Air Force had a program not too long ago called the Pilot's Associates Program, which gave some nice demonstration projects of the potential for this kind of technology. I think it reflected that the technology has progressed to the point where the Pilot's Associates Program certainly has a potential for impacting operations in a big way.

For our Desert Storm C-141 study, we were interested in whether fatigue affected performance. This is trend information, looking at the duration of duty day over a root mean square error of indicated airspeed over time. Once again, as the duty day for some of these pilots increased, the trend suggested some tendency toward approaches not being quite as good. The problem with these data is lack of control.

Air transport certainly has a long history of discovering, understanding, and illuminating factors to accidents. The technological growth for data collection processing, the FAA-mandated digital flight data recording systems, and engine condition monitoring programs have been around for a long time. But, they're not quite at the stage where they could necessarily predict the failure of a mode in a few minutes. I think we're getting at the stage where we can get some of that real time so the pilots have that information in flight, not just the engine maintainers on the ground who replace an engine. As an aircrew member, you never even know why that engine was replaced or what they saw, but occasionally you have in-flight failures that could possibly be predicted.

The thing that I think is more exciting is that we might be able to predict the human failures a little bit better. Pilot error is still cited in over 70% of hull loss accidents. (This is civilian data, now, and not U.S. Air Force data.) Takeoff, 19%; approach and landing, 39%. I think that we're getting to the stage where the capability of digital flight data to predict or prevent some of these pilot error type mishaps is certainly occurring.

The concerns that are expressed by the industry more concern the data security. How is data going to be used? And, how might it affect my career? The cost factors, as I mentioned, are coming down. There is a large problem with trust in the validation process in the validation of the parameters that you want to look at as far as: how do they reflect human performance versus other types of things that

BISSON: (Concl'd)

are going on in the aviation environment? It's a large problem there. And, how do you operationally interface the data that does occur with the crew member? The capability, I think, is easiest to interface after a training mission or something like that. However, there's also the opportunity to interface some of those data immediately with the aircrew in real time in the aircraft to affect how they are performing.

The operational type events you want to capture certainly include things like indicated airspeed, pitch, roll, and vertical velocity. However, the embedded performance task – the human performance task – something as simple as switching a radio frequency or dialing up a radar scope, detecting an object or something like that, these embedded performance tasks are very natural tasks for looking at response time, accuracy, and speed. All these computer tasks that we presently use as tools to look at human performance are embedded tasks in an aircraft. If you're familiar with running a checklist, it should take so much time to run a checklist; there are so many switches that need to be thrown in so many positions. Well, how many mistakes occur during that process? Something in the background can actually start capturing that. Something in the background monitoring that, it may be as simple as it's a design problem with the radio. It may be that the aircrew, in this case, is more fatigued. It may be they're missing radio calls. How are the human factors affecting these embedded performance tasks that we now do have the capability to look at?

The operational requirements for using digital flight data in the mode that I've been talking about is a method of looking at pilot and crew – talking about multicrew airplane – in-flight performance, looking at those interactions, validating those parameters. How do you best use them for training and for flight safety and for looking at aircraft trends?

You can use it for self-assessment improvement. Airport departure/approach design may be something you can do. Looking at digital flight data can also affect the way we design aircraft.

Those are some of the directions we're going in. It's an exciting field right now. The FAA is interested in pursuing this technology. Within the themes of this meeting and looking for how we can combine our efforts and join our efforts in transitioning technology between the civilian and the military sectors, I think this is just one exciting area.

KRISHEN: Did I understand you right, you said there may be some technology for planes that is used overseas but that we haven't used in this country. Is that true?

BISSON: The question was whether or not I had stated that overseas some of this technology is in use, but we are not using that this extensively in this country. Twenty-five of the non-U.S. carriers (and there's been a recent FAA report that describes some of the use in the European carriers – primarily – as well as, I think, JAL and some of those carriers), they are using digital live data routinely for monitoring trends and doing some of the safety and work that we've talked about. In the U.S. industry, there's not a single major U.S. carrier that I know of that routinely captures any of these data.

[The question is,] To what extent have we looked at AI technology in analyzing the data? There are a couple of programs that have been written. The software to interpret the digital flight data that exists on the commercial carriers exists, and so it's there. But, it's not in the artificial intelligence realm yet. It's more in just, How do you interpret the data? That's something that I visualize as a future thing.

The question was, Does 100% monitoring have its own cost in terms of stress, job satisfaction, and whether or not the European countries were including this in their analysis. I suppose I can't answer that directly because I have not looked at the European operations other than superficially, so I'm not aware of any studies that have looked at those sorts of questions. I know that, where it is being used, it seems to have been well accepted. The protective measures to make sure that it's not used as a punitive tool as much as a training tool and things like that are in place. And, certainly there are ways to make sure that, in at least some of the carriers I know, it's gathered in an anonymous fashion so that they may look at information but it would not be tied to a given individual.

ALEXANDER: The spread and the theme of this conference – as has been for all of our existence – is interdependency. I think it is probably never more obvious than in our next speaker how the work this gentleman has done in support of the NASA civilian program really is of keen interest to all of us because, even though the mechanical task may be somewhat different from what we do in the U.S. Air Force in our missions, certainly the physiology of man is the same; certainly the stresses placed on man are of interest to all of us.

Richard Hieb joined NASA back in 1979 after a graduate program at Boulder. He was with the Agency, at the Johnson Space Center here, for a few years, after being picked for astronaut duty back in 1985. He joined the ranks in 1986 and has two space flights under his belt in 1991 and 1992. Mr. Hieb helped us out with the DOD mission back in his first trip, and he had experience both with a free-flying satellite as well as with controlling the satellite in the payload bay

ALEXANDER: (Concl'd)

using the RMS (remote manipulator system) and then operations from the flight deck. His second flight resulted in a lot of records being broken.

Mr. Hieb's going to fly again, probably in 1994 in the summertime, on IML-2 and will continue to rack up his operational experience.

HIEB:

What I'm going to talk about this morning is my most recent flight because I think we learned a lot of operational lessons.

Now, I've got to say right off the bat that some of the old-timers will say – whenever I say some particular thing, somebody's going to say – "You know, we learned that back on Skylab." And, I suppose that's a lesson, too, isn't it? Because a lot of the lessons we learned somehow we managed to forget them over the years. Then we suddenly relearn them, and I guess that's not all bad. I hope someday, though, we'll figure out a way to keep those lessons learned and really keep our minds fresh on them.

Let me start up the movie because I'm going to talk from it.

We're kind of along for the ride for the first few minutes. You can read your checklist if you concentrate on it. But, for that first couple of minutes it's kind of hard to think about reading your checklist anyway.

When the SRBs come off, that's a great feeling. Like anybody else, we have these milestones. Everybody attaches different significance to different milestones. Certainly *Challenger* has caused us to add a lot more significance to getting rid of the SRBs.

This is the crew: Dan Brandenstein, Kevin Chilton, Bruce Melnick, Pierre Thuot, Kathy Thornton, and Tom Akers.

Exercise physiology. There's a lot we could talk about, but notice one thing: Kevin's got the ergometer stretched out in bungees because we're trying to isolate it from the rest of the Orbiter and reduce the vibration inputs.

Pierre and I are putting our long underwear on. These things are liquid cooling garments that have little tubes of water in them running through the suit.

We don't have any heating in our suit. We just have cooling. So when you want to get warmer, you turn off your cooling, and then your body Btu's will warm you up. There's a lot of thermal inertia in the suit. You have to kind of lead the

HIEB: (Cont'd) transient when you're going to go into shadow. You've got to turn your cooling down ahead of time.

Here's the Intelsat as we're coming up on it.

Pierre and I were busy getting ready as Dan's flying from back here. Here's an operational mess if you ever saw one. Kathy is operating the laser handheld, trying to get range marks. So she's handholding a laser. Bruce is looking out, getting ready to operate the arm. Pierre's riding at the end of the arm. He's got this long capture bar. This is the first try on the first day. And, he had trouble being smooth. The biggest lessons we learned on this flight had to do with our simulations.

I don't really hold to the story that there was a problem with the capture bar. The capture bar did exactly its job. It worked perfectly on the ground. Our problem was really in our simulation. Our simulation was not an accurate representation of what we were going to see in space flight. When Pierre went to put the capture bar on, the real satellite behaved a lot differently from the one on the ground. The one on the ground, the air-bearing floor, had some friction in it and it was a lot more stable than the real free flyer.

I'm not so sure that in fact maybe the best thing that ever happened to us on this flight was that Pierre was unable to make this capture bar attach to the satellite. Had he done that, it's not clear to me what would have happened next. We thought, based on our simulation, we had an idea of what our controllability was going to be. But, seeing how this thing was so squirrelly in flight, I'm not so sure that things wouldn't have been worse had he got the capture bar on.

I have to say, after the first day where we couldn't get a hold of the satellite, we were really down as a crew. After the second day, we felt better – sort of strangely. Although we still didn't get the satellite, after the second day we felt like we had gone and done our job really perfectly. We performed it exactly as we had trained on the ground, but we still didn't get the satellite.

A lot of discussion went on after the second failed attempt. Somebody came up with the idea of sending three people out to do a spacewalk, which doesn't seem revolutionary in retrospect but at the time it was.

The water tank is where we do our training. They went in the water tank while we were on orbit and figured out what was the right thing to do with three people and sent that plan back up to us.

HIEB: (Cont'd)

As we're waiting now, Dan is of course flying the Orbiter. The satellite has got a large coning angle. We've got three places we can grab, and we've only got a few feet of play that we can grab [those places] in. Our first plan was to grab it in the preferred orientation. Quickly, we realized that we were going to be lucky just to get any opportunity to grab all three handholds. So we said, "Okay, scrap the idea of grabbing the right three; let's just grab any three."

Honestly, after we got a hold of it, I thought to myself, "Well, I guess I didn't do any of the work." Because I didn't feel like I'd done anything to stop the satellite from rotating.

As it turned out, when we talked afterwards, none of us felt like we'd done anything. There's enough resistance in the suit that, in just moving our arms in the suit, there's enough overhead from moving the suit around that we really couldn't detect the force we were putting in trying to stop the satellite.

This was the part I was worried about the most. We've got the capture bar still to put on. We need to do that because that's what the arm has got to grab in order to control the satellite.

There were several operations where only two people were going to be holding on to the satellite. Our problem was that we had no good reference point. You'll see, as some of these pictures progress, the satellite starting to lean way over here. Tom could see it was a little bit crooked, but it was not obvious what we should do about it.

We had several minutes to hold it where there were just two of us. That was really the most tense part of this operation. We were out there for a couple of hours, holding on to the satellite, and ultimately that was what led to our setting records for the spacewalk, which certainly was not our intention. Ultimately, still as a crew we felt like we wanted to go up there and have everything work just like we trained to it.

At this point, Bruce has got it on the arm. The only problem was we're now about 3 or 4 hours into our spacewalk with 4 hours of work yet to do. We're clamping this thing on to a motor so that we can spring eject it out of the payload bay and get rid of it. We're working pretty quickly now because, already, Tom is running out of battery time.

But, after all this work of getting it, then we couldn't get rid of it. Somebody on the ground managed to get up the message that, somehow, the procedures

HIEB: (Cont'd)

reflected an old version of the drawings. The drawings that were correct showed that we had to throw the switches in a different way. They finally got up the corrected plan, and then we launched the thing and it was out of there.

If there are two things that we do for fun in space flight, one of them is taking pictures of the Earth. To look back down is a fantastic thing. I have to say, for those of you who haven't seen one of the IMAX movies, if you go see one, if they could project that onto a ceiling and let you float in a pool, you'd get a real good sense of what space flight is like. Because the view of the Earth in those movies is very, very representative of what you see looking out the window.

We used up three of our EVAs just getting the Intelsat in, but we still very much wanted to do some of the Space Station work. So we sent Kathy and Tom out to work on the Station stuff. This turned out to be much more difficult than we had really anticipated it to be. Some of these activities took way longer than they took on the ground.

In the water tank when you are working in a spacesuit, it's hard to get started, hard to continue moving, and certainly very hard to swing something like that big old pole around, because you've got so much drag in the water that you're always pushing on it. In space without any drag, it's easy to get things started. You don't have to do anything to keep them going. But, something that's different from the water is, you have to do something to stop them. Likewise when you're moving along in a spacesuit and you've got a couple of hundred pounds on your back, as you're moving along the slide wire it's easy to get yourself moving. If you're not careful, you get to moving too fast and then when it's time to stop, you grab on like you did in the water tank – except now you've got all this mass and momentum moving.

In the first half hour to an hour, you're very comfortable with how you move around with your body in a spacesuit outside. But, a number of the jobs that we trained to in the water tank just weren't the same outside.

For Space Station we're not going to have a rescue vehicle that somebody can quickly hop in. If a spacewalker gets cut away from the Station and goes drifting off, he's on his own. In a Space Shuttle, you know the pilot will probably fly after you and scoop you up in the payload bay. But with Station, you're a free-flying satellite. We had an idea of what the right alternate plan was for self-rescue, but there were a number of cheaper options that people wanted to consider on this flight. We did our absolute best to try and be objective and evaluate these things.

HIEB: (Cont'd)

It took me about 3 days to get used to being in zero gravity on my first flight. It took me about 3 days to get used to being back on the ground afterwards. My second flight was exactly a year and a day after the end of my first flight. I felt good within 1 day in zero g, and I reacclimated within 1 day coming back. So clearly there's some memory.

Let me go to the audience and see if there are some questions I can address.

The question was: With three folks did I feel like we could have manipulated the satellite any way we wanted to? I think absolutely. We could have done anything, but our control system was the guys inside the Orbiter. It was very hard for us to tell what we were doing. We could stop it from moving any direction. We went very, very, very slowly because we were afraid we'd get up to rates that we couldn't handle. So we did everything extremely slowly. But, we really couldn't see it.

Let me comment on the question that this lady up front asked – the last speaker – which was, "What does it feel like? Does it add stress to be monitored 100% of the time?" When you're doing a spacewalk, you are always monitored 100% of the time that there's ground coverage. So everything we say goes not only inside the Orbiter but to the ground. For the rendezvous, Dan wanted to do something unusual; he wanted to put the Orbiter crew on hot mike as well. So that everything they said from the time he took over to start flying manually, everything they said, everything we said would be available to the ground. I have to say that, during the times we were operating where I knew that we could be listened to, I didn't think about it. I don't think it affected me greatly. But on the other hand, whenever we knew we were LOS and there was nobody going to be listening in, there was a definite sense of relaxation. So I would say that I think, even as much as we're used to it, clearly it's something you do have to get used to and something that's nice.

Preparing to come home. We did a lot of different things. The question was about fluid loading specifically. For those of you who aren't familiar with that, we lose a lot of fluids in space flight because, when you go to zero gravity, the fluids tend to shift upwards. There's no gravity pulling the fluids down there. When it's time to come home, if you haven't done anything to prepare for it, that fluid goes right back down to the legs where it wants to be in one g and now there's not enough for your brain. So there's a threat as to whether or not you're going to be conscious and able to do all the things you need to do when you get back to one g.

HIEB: (Concl'd)

To counteract that we've tried a number of different things. The current countermeasures that we take are: we take a couple of salt tablets with every 8 oz of water, and we're supposed to drink a minimum of 32 oz of water. There's a drug called Florinef, which we're testing on a few scattered subjects, trying to find out if that will help us retain water.

Let me get back to one thing. Simulation, it's the one thing I want to focus on in operations. Simulations are incredibly important. We depend on them. That's all we do here. It seems to work. But you have got to know, you just have to know, what the limitations of the SIMs are.

The water tank? Yes, we know about water drag; but we didn't focus on it as hard as we should have. The air-bearing floor? Some people knew there was friction in that air-bearing floor. We did not fully appreciate how much friction there was, because the real satellite didn't act like the air-bearing floor.

The simulators are great things. We learned a lot using them. But, if you don't know where the holes are, you're going to be in big, big trouble when you depend on it.

ALEXANDER:

Our next speaker is Dr. Howard Schneider. Howie's position is one of defending the scientific investigation in the engineering and the other objectives of any space flight mission. Howie leads our Experiments Review Board. He represents the scientific point of view during mission operations, and he pretty much has the final say as to what goes and what doesn't go. Howie's well trained, with a Ph.D. in biophysics and a Masters in biochemistry out of the University of Houston.

SCHNEIDER:

I'm going to talk about science operations instead of science. The operation is how we get there. These are not textbooks, and I don't ever plan for them to be in there. You have output and input. And, you have a human/simple machine interaction that's linear. Here you have a computer. A human with small input with a very large output from the computer.

As you see there, we don't know what in the world's going to happen a lot of times in science. When we start, we've got a group of people we're monitoring 150 miles in the air and we see them every 45 minutes above us.

What drives this science OPS? Well, as far as the science goals that we have in life sciences, it's not a perfect science, as one knows, because humans are involved in it. We want to ensure the health, wellbeing, and productivity of

SCHNEIDER: (Cont'd)

humans in space. We want to develop an understanding of the role of gravity of living systems. You've heard in the previous talk some of the things they've experienced firsthand.

We want to expand our understanding of the origin, evolution, and distribution of life in the universe. I've often wondered where I came from. Maybe I don't want to know, but that's one of our goals. Certainly one of the most important in these days, as Dr. Cohen said, in going up and talking to Congress, we want to promote these applications of our life sciences research and promote the quality of life here on Earth. To that end, we try to take these findings that we have from space flight and actually apply them to the everyday life of people.

Let me show you how this starts out and how we have questions about the physiology of space. What happens? We go to the outside community, put out an announcement of opportunity, and say, "Can you help us with this?" We had over 400 proposals that were submitted for Spacelab-4. In our naivete in the operations, we thought we could do a lot more than we could. We thought we could have people working 26 hours a day and do everything. But, we found out we were wrong. We split that into two missions, and then we called it Spacelab Life Sciences-1, which flew 2 years ago, and Spacelab Life Sciences-2, which will fly in September.

Spacelab-1 was extremely successful. Many of the experiments that we flew in SLS-1 we will repeat next month with some enhancements from lessons learned. And, we will provide a larger sample population. As you know, our samples are very few. The subjects that we have are the crew people.

We had 10 investigations on SLS-1 using humans. Four of them were cardiovascular, cardiopulmonary; two were musculoskeletal; three were regulatory physiology; and one neuroscience. Again I'd like to talk about operations. We have principal investigators sitting off in their institutions. So what we do is collect data for them in an orderly fashion and make sure that they get it in a pristine fashion so they can then in turn make assessments of what really happened.

This is Millie Fulford-Hughes, a payload specialist; and this is Jim Bagian on a rotating chair. They're patched up and wired up. They want to study the space motion sickness that you hear about and also the vestibular ocular disturbances. They'll spin that around. Eventually Jim will take his head and try to dump

SCHNEIDER: (Cont'd)

what they call nystagmus. All that time we're collecting data. We have one chance to get it, and that's the science operations part of this.

On SLS-1 we took some rodents along as passengers. They had essentially the same group of experiments to perform pre- and postflight on these animals. We did some hardware verification of the Research Animal Holding Facility and the General Purpose Work Station in which we can handle mildly toxic chemicals and manipulate the animals and do some procedures that are impossible to do on humans.

We look very well after these animals. We have an Animal Use Committee – it's much like our human Policies and Procedures Committee – that makes sure the animals are treated humanely. And, it protects the astronauts from the animals (from cross-contamination) – and also the animals from the astronauts, so they don't cross-contaminate one another.

To talk about space flight, we have a limited number of opportunities for flights. Crews on some of the other flights, who are not fully dedicated to life sciences, do experiments – as you heard, the lower body negative pressure, which is an OPS type thing. But SLS-1 and SLS-2 are dedicated more to the basic science than they are to operations. We have a small subject population. We have a limited number of samples. These samples are as precious to the life scientists as the lunar samples were back in 1969 to the geologists.

There are seven people in our crews: three who maintain the Orbiter and four who are responsible for the science operations in the Spacelab. The Orbiter crew is more than generous with their time and help us to far exceed what we expect to get under the normal circumstances when they volunteer to be subjects and participate as operatives also.

Lest we forget, this is the SLS-1 crew. Jim Bagian, Sid Gutierrez, Drew Gaffnay, Bryan O'Connor (who is now at Headquarters), Tammy Jernigan, Rhea Seddon, and Millie Fulford-Hughes.

Continuing on the resources of the space experiments in life sciences. The crew time is limited. The experiments have to function within the resources.

I'd like to talk about the use of animals. They're going to fly on SLS-2. I'm sure that there'll be a lot of press about these animals and what's happening to them. Certainly, we are prepared here at NASA and NIH to have rational ideas as to why we need to do this in space. We want to validate them for human models.

SCHNEIDER: (Cont'd)

We have a larger subject population. Again as I talked, even at the end of SLS-2, we would have six humans on most of the experiments; and that's not very much. We can fly 24 rodents in each RAHF, for a total of 48 rodents. You recall the jellyfish, possibly, on SLS-1. There were 2478 jellyfish; they got a lot of press.

Part of operations is to try to schedule activities to where we don't duplicate data. It always takes longer to do something in zero gravity than it did on Earth. We have a shopping list of experiments available each day for the crew. They know what they can do. They know how tired they might be. They know how they feel about doing something. If for one reason, one of our experiments doesn't seem to work that day – or if they end up with extra time – they can go on and do this.

Communications: As part of our science operations, we communicate with the investigators and to the crew. We have a science operations planning group that meets every night during the mission, and we poll all of the investigators where they can express issues concerns. We report daily accomplishments, be they good or bad. And then, with all of this information, we review and replan our next day. Via the air-to-ground communications with the crew, we track the status and we are prepared then to respond to any anomalies.

We go through the TDRS system, through White Sands. The POCC that performs the operations for Spacelab is located at Marshall. Certainly at JSC you have Mission Control, and then also at JSC you have the science monitoring area. Here is where they monitor the human experiments, and the PIs reside there and can communicate with crew members. Those with animals are at KSC or Dryden.

Our hardware is a different kind of hardware than most people are used to. We have things like refrigerator/freezers and a Urine Monitoring System (UMS), the Gas Analyzer Mass Spectrometer, and a backup echocardiogram.

We do try to plan and have contingency plans, looking for anything that can go wrong and to see how we would handle that. We develop detailed contingency plans for these samples that I mentioned were so precious so we can save them in case there's a loss of power, loss of a refrigerator/freezer, or loss of an instrument. We document where each sample is located. We track and update during the mission where they are and the condition of those samples through crew interaction. And, we establish priorities prior to the mission that establish how we can best get the most science out of this one shot that we have.

SCHNEIDER: (Concl'd)

This is a refrigerator/freezer; and during the mission, these behave differently in zero g than they did on the ground. They can only have but just a few ounces of Freon. So they have a high compressor rate.

As you know after the Shuttle lands, 98% of the people are gone. People like myself remain around. We have to get all the final reports in. We have experiment data (reams of it) and in-flight photography. Much of this is private medical data, so we have to be very careful about how it's released to the public or the Press. There's a 30-day quick-look report that we put out, and that's to see if there's anything that really hit them square in the face that we ought to talk about that they can use, maybe, on that next coming mission. Any person who is in science doesn't like to give you a 30-day report unless there's something in there that they really see you ought to know about. The 180-day report? We talked about Congress; I think it's a political report. We get that ready so we can send it up to Headquarters. We try to find out something that was scientifically important, and we send that up for them to help with our budget. Then there's a 1-year contractual agreement.

In summary what we do is, we try to integrate the investigations to share hardware, protocols, and samples wherever possible. We develop contingency guidelines and plans and procedures. And, we establish all the priorities for the science for the payload elements prior to the flight; hopefully, that works in the flight. (Obviously, these are motherhood statements.) We apply the knowledge gained from each mission to the next mission, and establish a plan for postflight data dissemination and reporting.

MURATORE:

What I want to talk to you about today is not about the technologies per se, but to give you a flavor of operationally what we're able to do and the kind of problems we're facing in the Shuttle Program.

First thing I'm going to talk about is the big picture, in order to give you kind of a perspective. Space Shuttle is now the major part of our experience in human space flight in the United States. We've flown 57 missions over 12 years, and we've got over a full year of on-orbit time accumulated. Prior to Shuttle, there had only been 29 piloted missions in the United States space program. So really Shuttle represents the bulk of our experience. If you select the metric of numbers of payloads to orbit or the numbers of pounds of payload to orbit, the Shuttle Program is by far the most productive in the United States space program. Certainly, you could select other metrics, but just taking those two we've carried more things and more pounds of things to orbit to do useful things than any other manned program.

MURATORE: (Cont'd)

We've demonstrated a surge capacity of one flight per month, and we're maintaining an average flight rate pretty successfully of seven to eight flights per year. I think that the important thing is we're maintaining that flight rate while at the same time lowering the operations costs 5% per year. I think it's very important for NASA because, in order for us to go and attack other challenges, we have to find ways of bringing the operations costs down so we can build wedges to go invest in the next generation of activity.

When I talk about Mission Operations and I talk outside of the Mission Operations community, often there's a little confusion as to what is Mission Operations. I'm talking about the whole package. It's not just the people sitting on the console. It includes facility development and maintenance for our control centers, our simulators, and a lot of off-line facilities that we need to make the missions run. The flight design aspects are to design the trajectories, managing and planning the consumables, planning what we're going to do with the robotics systems. It involves reconfiguration – and reconfiguration is taking the software loads, the basic software capabilities, and tuning them for a specific flight profile. We build the control center and the Shuttle mission simulator. We have to do flight planning and procedures development. We do crew and flight controller training as part of the Mission Operations function. Then finally, we do the flight operations part – the part you see – monitoring and controlling of the systems, trajectory, and payloads.

Most of the mission resources are spent in facilities development, maintenance, flight design, and reconfiguration far more than actually in flights – direct flight – support. Most of that cost involves software maintenance and operations. There's tremendous opportunity there to use advanced technologies to drive cost down. Most of our people are involved in maintaining software today. Not only do we need software to automate the functions, we need better ways of maintaining and managing that software. When you come down to it, the actual people on consoles are a very small part of the total mission operation.

In flight operations, we work the consoles 24 hours a day, 7 days a week during the missions. Based on the flight phase – a little more for ascent and entry; a little less for orbit – between 50 and 80 people per shift are responsible for monitoring and controlling the Shuttle systems, the payloads, the trajectory and flight planning.

This number is reduced significantly from the original numbers in early Shuttle flights, even though we've added things that weren't in the early Shuttle flights. We've added extensive RMS activities, the EVA, rendezvous, and things like

MURATORE: (Cont'd)

that. Based on what's going on again in the flight, it may take double that size of a team to keep the facilities in the buildings running – somewhere in the order of 80 to 100 people per shift. Keeping the control center systems operational; keeping the command system up and running, the voice systems, TV, network, computing, electrical power, air conditioning, security, those sorts of functions. There's another real opportunity for automation and for advanced technology.

Five to six teams are necessary for us to fly 8 to 12 flights a year, and that's because we tend to clump these missions up. We've done a lot of work with expert systems in the control center; and, in fact, in the next generation control center, which we're in the process of delivering right now, we are making expert systems and advanced automation a baseline part of the basic technology.

Where are we today in August 1993 with our experience? We've had 57 flights. And, a test flight program of 57 flights would be considered extremely short for a new commercial or military aircraft. The YF-22A, the next-generation U.S. Air Force fighter that's been selected, was based on a demonstration and validation program. The X-29 in its basic flight test program accumulated over 200 flights, and then it extended beyond that. Our ascent and entry experience base is 57 flights.

Our on-orbit experience base is large in comparison to the others. Like I said earlier, we have almost a year accumulated on orbit. So the conclusions and the kind of steps we can take in orbit are very different from those we can take in ascent/entry.

Let's talk a little bit about ascent/entry. The environment remains very challenging, and there's still a lot to learn about flying ascent/entry. The Orbiter's guidance, navigation, and control system uses the drag it gets through the atmospheric density to help it compute its altitude and how it maneuvers to maintain enough energy to make it to the runway. There are going to be surprises like this (I bet) through the next 50 flights. Not many people fly up where the Shuttle flies.

What we have to do is maintain margin both in the operations and in the teams to make sure we have that extra propellant when we need it, to make sure we've got that extra structural factor of safety, and we've got the teams trained to very high performance standards.

For ascent/entry, we need to continue to maintain the margin. Now we can maintain the margin by keeping the same trained teams in there. We can

MURATORE: (Cont'd)

maintain a margin by having structural margins and propellant margins, things like that. We can also maintain the margin by having the automation techniques in there which allow us to operate with the same level of margin with less people or with less time.

On orbit we found that the Shuttle's systems perform really well with a near zero rate of anomalies on orbit. It's a real testimony to the excellent work of people at KSC and of the care and expertise that the design community put into it in initially designing the Orbiter. We're going to be flying new TV cameras on the flight in December. We've demonstrated the capability to integrate upgrades into the fleet without interrupting the flight schedules. This list here is a list of all the things we've added since STS-26 in September 1988. All of these things we're flying with today that are new and upgraded we did not have in the Orbiter when we lifted off on STS-26. [Among these are] new onboard computers; mass memories, which are the tape units; new inertial measurement units; new star trackers; TACANs or radio navigation system; new fuel cells; new power units for the hydraulics; the new printer; new waste collection system. They've all been integrated into the fleet since return to flight.

Our next big activity is upgrading the Orbiter to a glass cockpit. That's a pretty big step when you consider not only do you have to upgrade the Orbiters but also the simulators and the flight software test facilities, and you've got to integrate that all with the flight software changes.

As it turns out today, most of our on-orbit operations time is dedicated to payload and experiment operations. We've managed to spend very little of our time flying the Orbiter and taking care of the Orbiter systems. We spend most of our time doing payloads and experiments.

The Shuttle as a system places relatively few constraints on payload operations. We're concerned about safety issues. But, if it's a payload that is compatible in any way with Shuttle, usually the requirements placed on the payload are pretty small. We have shown through a lot of instrumentation we've flown on a number of flights that we provide a very good microgravity environment. We have shown that we're an excellent pointing platform for whether you're looking at Earth, at the sky, or at other objects in orbit. We do have constraints to payload operations. However, most of these constraints tend to be between multiple payloads on the same flight.

The typical integration issues are the kinds of things here. Everything from attitude and pointing to power outlets to stowage. We can lift more in the Shuttle

MURATORE: (Cont'd)

than we can do with the crew for the missions we've got. We timeline the crews very heavily. We deal with problems when we get on orbit and get a lot of science work out of them. We generally can carry more weight in terms of experiment than we've got time with the crew to do the work. Almost all of their time is spent working payloads and experiments. Very little of it is spent taking care of the Shuttle.

In-flight maintenance: The big concern on Station has not shown to be a bigger driver on Shuttle missions. Most of the IFMs are there not as safety of flight issues but they are there in order to improve mission success.

The Tracking Data and Relay Satellite network has shown itself to be very reliable. We use it heavily, and we capture huge amounts of scientific data. However, what we're finding is the network's very busy – not only supporting us but Hubble and Lansat and other users. We're having to constantly adjust our usage of the satellite in order to give other users a chance.

The remote manipulator system is a very mature system with tremendous capabilities. We use the arm in practically everything we do. We flew an experiment on STS-52 called Space Vision System. The Space Vision System took in video and then analyzed the video and gave us synthetic views of the payload and also gave us all sorts of information about range and range rate and attitude that were a major assist to the crew working the experiments. That that's the kind of capability that's really going to take us to doing our next big step and capability.

EVAs: Development in the EVA area is a continuing activity. We learned a lot on STS-49, and we've readdressed the way we do EVAs. We've made really big improvements in the system – the entire work system; not just the suit but also the tools that we use, the platforms, the restraints, and in the training in the last year. We have instituted a regular program of EVA flight tests.

The last thing I'm going to talk to you about is the impact of personal computing in Shuttle operations. We are finding that the personal computers are becoming a major on-orbit tool.

They have off-line uses, too. For Shuttle, we used to fly calculators in case the ground went away and you had to compute your own weight and c.g. for entry and things. We had little calculator programs to do that, and now all that's in the personal computers. We're using them quite a bit for real-time computing, with real-time interfaces to payloads. During the Tethersat flight, we actively

MURATORE: (Cont'd)

controlled and monitored the Tethersat using a personal computer sitting in the aft flight deck. The laser range devices were really important.

On this next flight, we're going to demonstrate real-time monitoring of Shuttle telemetry with onboard personal computing. We had a port on the Orbiter where we could attach into the real-time telemetry what the Orbiter was transmitting down to the ground. Jim Newman, who's one of the crewmen on STS-51, led this charge of attaching the computer to the telemetry so that we can have real-time displays not only of the Shuttle's data but of payload data on the personal computers.

We've been doing a lot of work linking the personal computers on board to the computers on the ground using modems, sending tones over the air-to-ground voice loops. That has been very successful. We had a SIM debrief on STS-56. You heard the crews talking about using personal computing using the same tools they have in the office.

The crews have been working on taking observations and working on their postflight reports in-flight. They take the computers with them into quarantine and take a disk with them on to the Shuttle when they fly, and they just kind of keep a running report into our flight. We're trying to merge doing functions of hard real-time computing with office type automation functions.

So that gives you a picture of where we are in terms of Shuttle operations today and of the kinds of experiences and the kinds of trends that are under way. I'd be happy to answer any questions.

Glass cockpit. That is replacing all the traditional electromechanical flight instruments with CRT displays – usually color – so that you can go ahead and, instead of having to maintain these electromechanical displays, attitude indicators we call horizontal situations indicator will tell you your relationship relative to the runway. Tapes that indicate your velocity. We're replacing all of those with a CRT electronic system. During ascent most of them are changing so fast that they're not very useful. It gives you the option of changing and then integrating things like graphics into the display formats.

Yes, it was developed by the Canadians under the Canadian National Research Council. What it did was, on the Canex experiment on STS-52, which was a small payload on the end of the arm, it had small reflected targets on it. What the system could do is take in video through the Orbiter's video system and then, by looking at the targets, do a lot of math very quickly to compute distance and

MURATORE: (Concl'd) attitude of the payload. As you could see from the videos of the Intelsat work that Rick showed early in the day, it's not at all clear what the attitude or attitude rates on the vehicles are. It's very hard to determine that.

So if you can do it by just looking at the object, that's a tremendous capability. It had onboard processing, which displayed to the crew a synthetic image along with visual data, and then also we sent the video to the ground and we ran it in parallel on the ground. It's a tremendous potential area for improvement and it really can make a big difference. I think it's the next big step in robotics in space.

ALEXANDER: Our final speaker this morning is U.S. Air Force Maj. Mark Pestana. Mark is with the Space and Missile Systems Center, which is headquartered at Los Angeles, California.

Mark really represents what interdependency is all about. His topic today will take technologies that are appropriate and relative to the NASA mission and apply them to relevancies and realities in the U.S. Air Force reconnaissance mission. Mark's had experience in the Mountain Shine mountain up in Colorado tracking our space assets and mission operations there. Mark has had some experience at Vandenberg in working in space tests and had experience in space flight experiments, looking at unmanned launch vehicle. He's a command pilot, a KC-135/RC-135 configuration.

PESTANA: I'm going to talk about how the application of space technology has improved an operational system that's been in existence for quite a while that I have had personal involvement with. That's flying the RC-135 worldwide reconnaissance missions.

First I'll describe what an RC-135 is and what the mission is about. Of course a lot of it remains classified. There are certain requirements to accomplish the mission, and there are systems required to perform the mission that we're dependent on. There are limitations to performing that mission because of the systems. Then I'll talk about how the introduction of space technology has improved the overall effectiveness, efficiency, and safety of the mission.

The RC-135 is basically a Boeing 707-type airframe. The RC-135 is especially modified for electronic reconnaissance. Right now there are five versions flying, and they're designated by a suffix at the end of the designation. So there's the RC-135U model, and there are a V, a W, an S, and an X model.

PESTANA: (Cont'd)

Each one has different capabilities, and each is used in different capacities. The home base is at Offutt Air Force Base in Omaha, Nebraska. However, to perform the mission you must travel overseas. There are five operating locations: Kadena, Okinawa; Shemya Air Base at the end of the Aleutian Islands; Mildenhall, England; and Iráklion on Crete, in the Mediterranean. There's an additional provisional unit that was activated 2 years ago in Saudi Arabia that's performing a mission over there.

Depending on the model of the aircraft, the RC-135 contains 33 to 35 crew members. Of course in the front of the aircraft are the pilot, copilot, and two navigators. The rest of the back end of the airplane is full of electronic warfare officers, systems specialists, and even a couple of in-flight maintenance technicians. The characteristics of this mission include the capability for long-duration flight. A typical mission is about 13 hours with one air refueling. The capability is there for extended duration – indefinitely – usually augmented by additional flight crew members so people don't get too fatigued. Of course during Desert Storm, we had RC-135s performing 24-hour coverage, so we had two to three airplanes in a shift.

These flights are flown over international waters – over friendly territory. It's flown overtly on a flight plan up to a certain point. It's not a stealthy aircraft or a fast aircraft, so it's not like we're surprising a lot of people. And, it's recognized by international law as a reconnaissance mission – not a spy mission. So I just want to emphasize how this is a legal activity that's recognized by international law.

The mission usually begins several hours after takeoff with a rendezvous with a tanker (a KC-135), where we take on fuel up to our maximum in-flight gross weight capability of about 300,000 lbs, and this allows us to fly this extended mission.

Of course the mission is to gather electronic intelligence. Those are subdivided into two categories: signals intelligence and communications intelligence. And, that's all I can talk about it.

The capability, as I talked about before, is to have 24-hour tasking with several aircraft assigned to a certain area. However, the routine operation is just a daily mission, wherever the specific area of interest is. Of course during times of crisis, like in Desert Storm, we can have aircraft cycled continuously for 24-hour a day coverage.

PESTANA: (Cont'd)

As I mentioned, the navigation accuracy: we're dependent on that to perform the mission and also to guarantee where we are so we don't violate air space. Threat assessment: we're a pretty vulnerable aircraft. We're an unarmed aircraft, virtually hundreds of miles from any friendly forces, and we'd like to know the status of the threat out there in relation to our personal wellbeing. We have people in the back end of the airplane who can provide that status through various means. We also have some other assets that can provide us status externally, and that's why we need the radios.

Some of the limitations to these systems: As far as HF communication, it uses the ionosphere to bounce the signal around the world. During periods of solar activity, it's subject to interference. The radios on board the airplane are also used to transmit collected data – especially items of high interest that need to get back to the end user immediately.

Talking on those radios is in the clear. There is no encryption capability. So we must manually look at code books when we need to send out certain messages, and we must encode these messages. Also, when we receive messages they're sending code and we must decode those. As Col. Bisson noted, a fatigue factor that affects crews on long missions: one I rank right up there with all the noise and vibration in the airplane is the nose on that HF radio.

Limitations to the navigation systems include: the INS over a period of time will tend to drift; that is, its determination of where we actually are begins to drift. We rely on a star tracker to give us updates, or the navigator can use his radar to take fixes off of land masses and determine our position. And then one can re-input those positions. However, in the polar regions this becomes very difficult because you can't take a fix off of ice. The coastline is very indeterminate in the polar regions.

Weather conditions are another problem. The high overcast in some areas that we fly does not allow for the star tracker to take reliable updates.

We've been able to improve our mission capability by introducing some space technology into the system. The aircraft is equipped with an AFSATCOM terminal. It's a keypad with a printer type operation. Now we have another communication pathway to use; multiple links are available. Messages can be stored – prestored – on the ground, certain critical messages that may need to be sent out during the flight. These messages are automatically encrypted. There's no requirement to encode them manually.

PESTANA: (Cont'd)

Aircraft are also now equipped with GPS receiving systems, a multiple satellite configuration under constellation that allows us to achieve highly accurate navigation capability. GPS applications add another level of redundancy in case we have some systems failures.

We've taken weather satellites for granted for so long that I have to include them here, because the missions of the RC-135s are performed in areas where we just can't rely on a weather station to call. Having satellite weather forecasting, weather observation capability is tremendous for us to know what it's going to be like when we get there – either for the area we're going to or our final destination.

There's a tremendous improvement in the overall effectiveness of the operation. We've been able to perform the mission with less aborts; that is, when systems fail, we have another system available to take over. We can improve our overall effectiveness in determining the intelligence picture – the threat – because we can precisely locate certain targets using the highly accurate navigation capabilities. Also, just performing the mission has improved. The ability to communicate better, easier, and also meet up with our tanker better to get that fuel to extend our mission. Overall safety has improved. We have better weather warning capability as well as timely threat assessment, threat warnings, via the AFSATCOM.

So in conclusion, I think there are two lessons learned here that are important to all of us. Lesson number one is that, as a user (as an operator), it would have been very helpful to know about improvements in technology that could apply to my operation. Likewise, we as developers have to recognize what operations are going on out there that require improvements.

That's all I have, unless you have some questions.

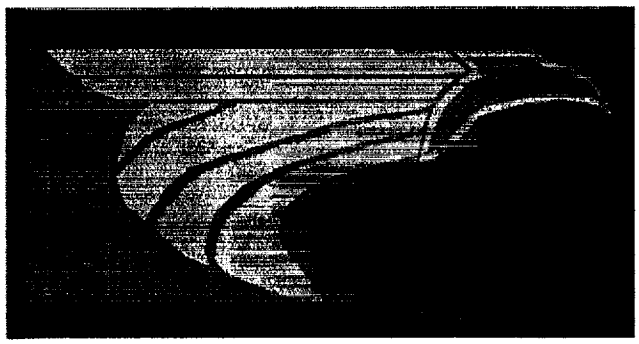
The question was about having GPS. Does GPS eliminate the need for having two navigators? No, because as I mentioned: it's another system, but we like to have redundancy. There's still a requirement to do two independent assessments of navigation. Even though the GPS is available, we'd like to have that backup capability. During wartime, especially, you don't know who's going to negate your capabilities; in other words, who's going to start shooting down some satellites. Although the threat has diminished, there's still a threat out there. And, we still like to have a capability for redundancy to perform the mission.

PESTANA: (Concl'd) As I speak here, these aircraft are still performing a daily mission worldwide of just kind of keeping an eye on what's going on. I think that's it.

ALEXANDER: Many of you heard me say last year: there are no unmanned systems. Man is an absolute integral part of everything that we do. It enables these more mechanical devices to reach their full potential. No question about that. Together we operate in concert for mission completion and mission success.

Ed. Note: Vugraphs relevant to the preceding presentations appear on the following pages. If you feel your organization would like to see the videotapes from which these transcripts were taken, please let us know.

SOAR - 1993



Johnson Space Center
Houston, TX
Aug 3, 1993

Performance Enhancement Initiatives Supporting

Global Reach - Global Power

Operations Experiences
SOAR - 1993
Johnson Space Center
Houston, TX

Dr Roger U. Bisson
Sustained Operations Branch
Armstrong Laboratory
Brooks AFB, TX
(210) 536-3464
Aug 1993

The ENEMY FATIGUE

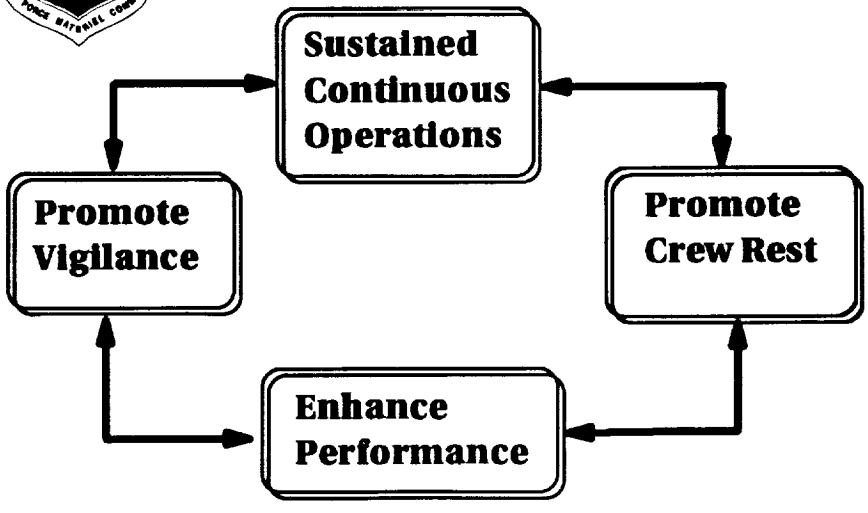
Aliases: Exhaustion, Weariness, Lassitude, Apathy, Languor, Burn Out

Intelligence Threat :
Errors in Judgment
in Accuracy
Decreased Response Time
Irritability
Blurred Vision
Muscular Weakness

THEATER OF OPERATIONS
- ★ SLEEP LOSS
- ★ CIRCADIAN DYSRHYTHMIA

Objective
- ★ Contain The Threat
- ★ Neutralize The Enemy

Fatigue Countermeasures



DELIVERABLE WEAPONS

Promote Vigilance
- Digital Flight Data
- Bright Lights
- Tyrosine
- Modafinil
- Exercise/Nutrition

Promote Crew Rest
- Restoril
- Melatonin
- Sleep-Wake Schedules
- Napping

CURRENT OPS

SUSTAINED OPERATIONS BRANCH
- Melatonin Quality of Sleep
- Shiftwork and Fatigue (ATC)
- Long Duration CONOPS (B-1B)
- Fighter Pilot Fatigue (CAP, Deployment)
- Desert Storm C-141 Fatigue Study
- Warbreaker
- Computer Model (Fatigue/Performance)
- Fatigue Management Doctrine

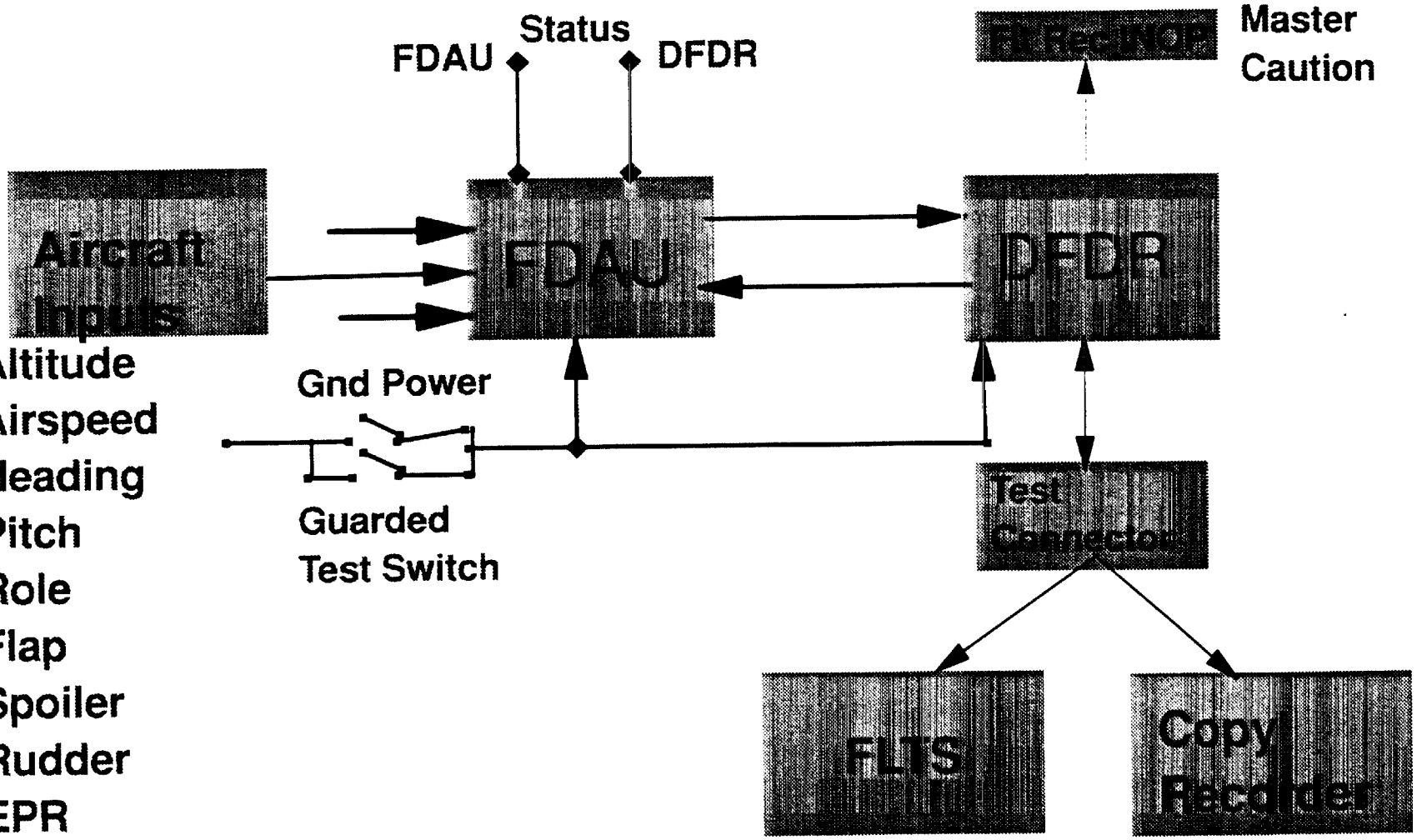
Digital Flight Data
A Defense Conversion Opportunity

- Operational Experiences
  - B-1B LaJunta, CO
  - B-1B Ellsworth AFB, SD
- Desert Storm
  - C-141 Fatigue Study
- Defense Conversion Opportunity
  - Historical Uses
  - Safety, Training, Trends

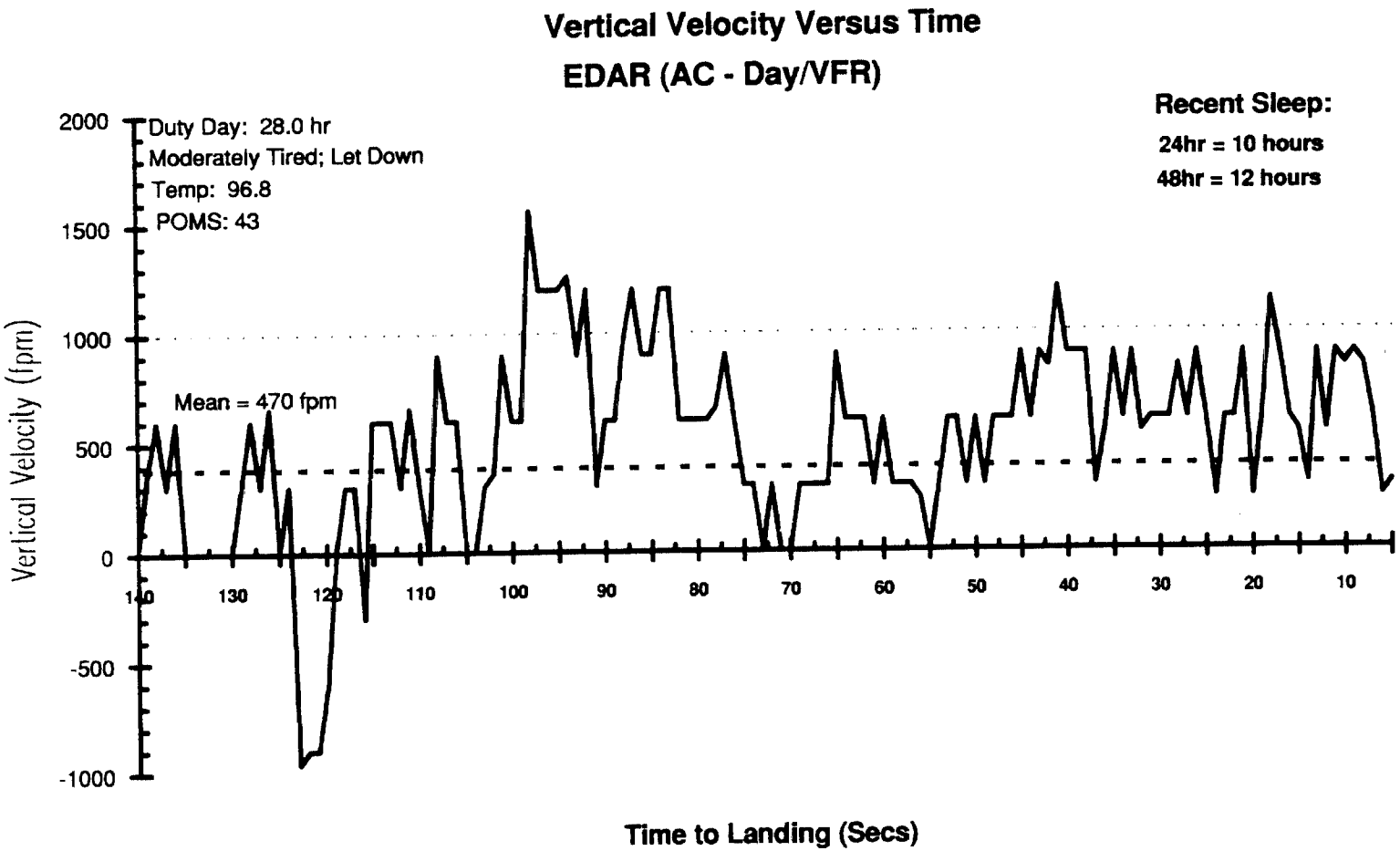
Digital Flight Data
A Defense Conversion Opportunity

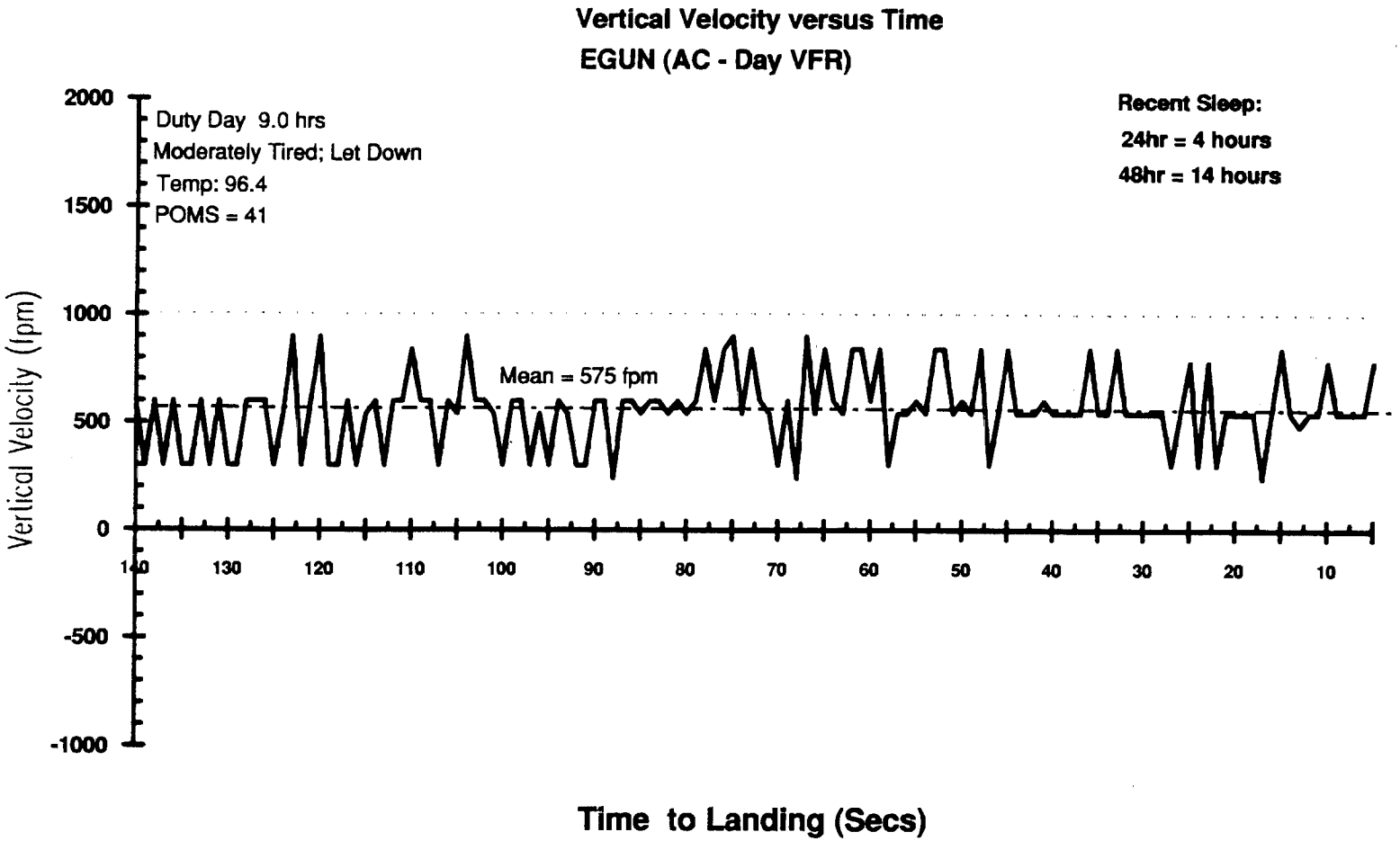
- Technological Growth
  - Data Collection and Processing
  - FAA Mandated DFDR systems
  - Engine Condition Monitoring
- Pilot "Error"
  - Over 70% of Hull Loss Accidents
  - Takeoff 19%; Approach/Landing 39%
- Flight Operational Quality Assurance

Digital Flight Data Recorder
System Diagram

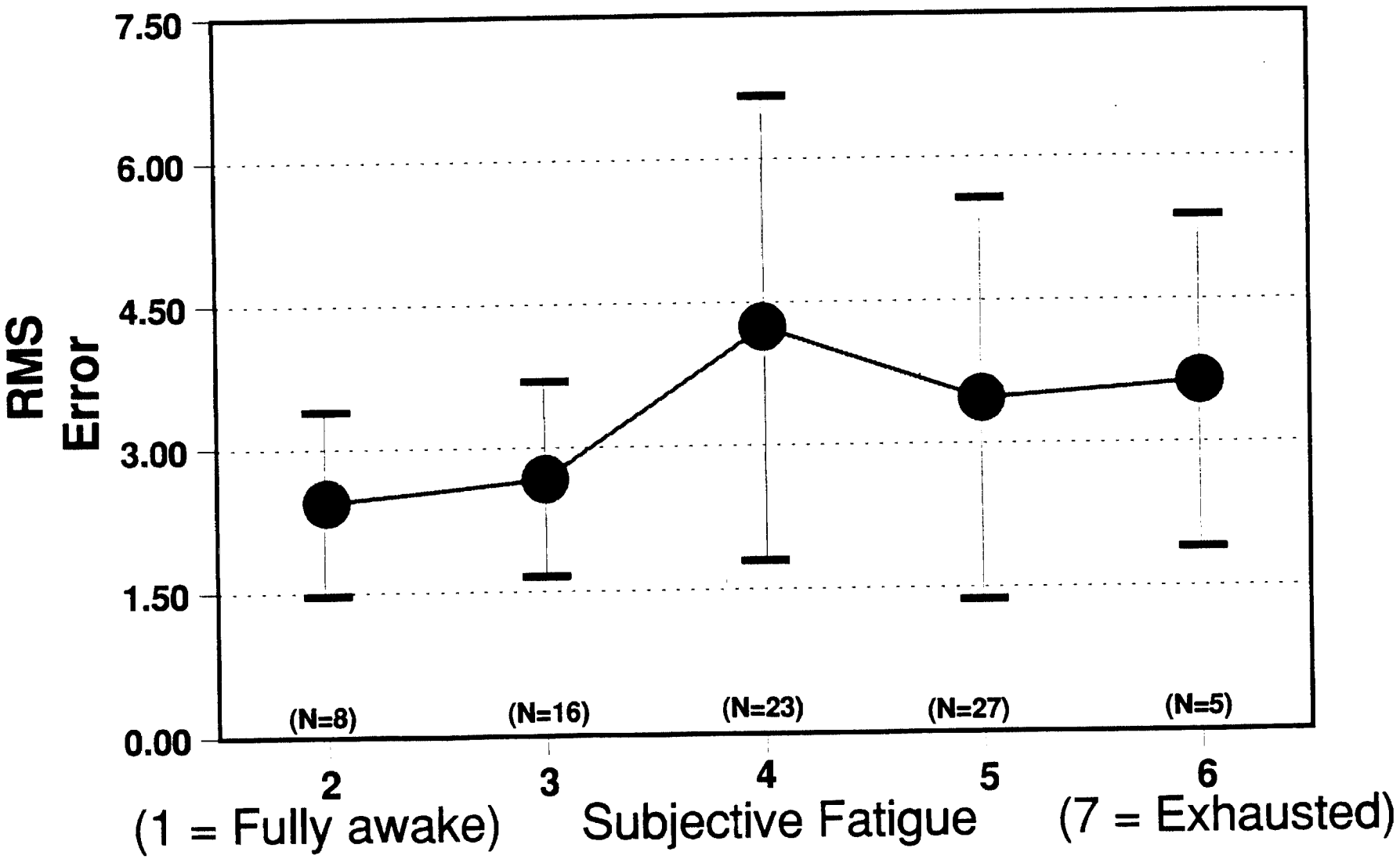


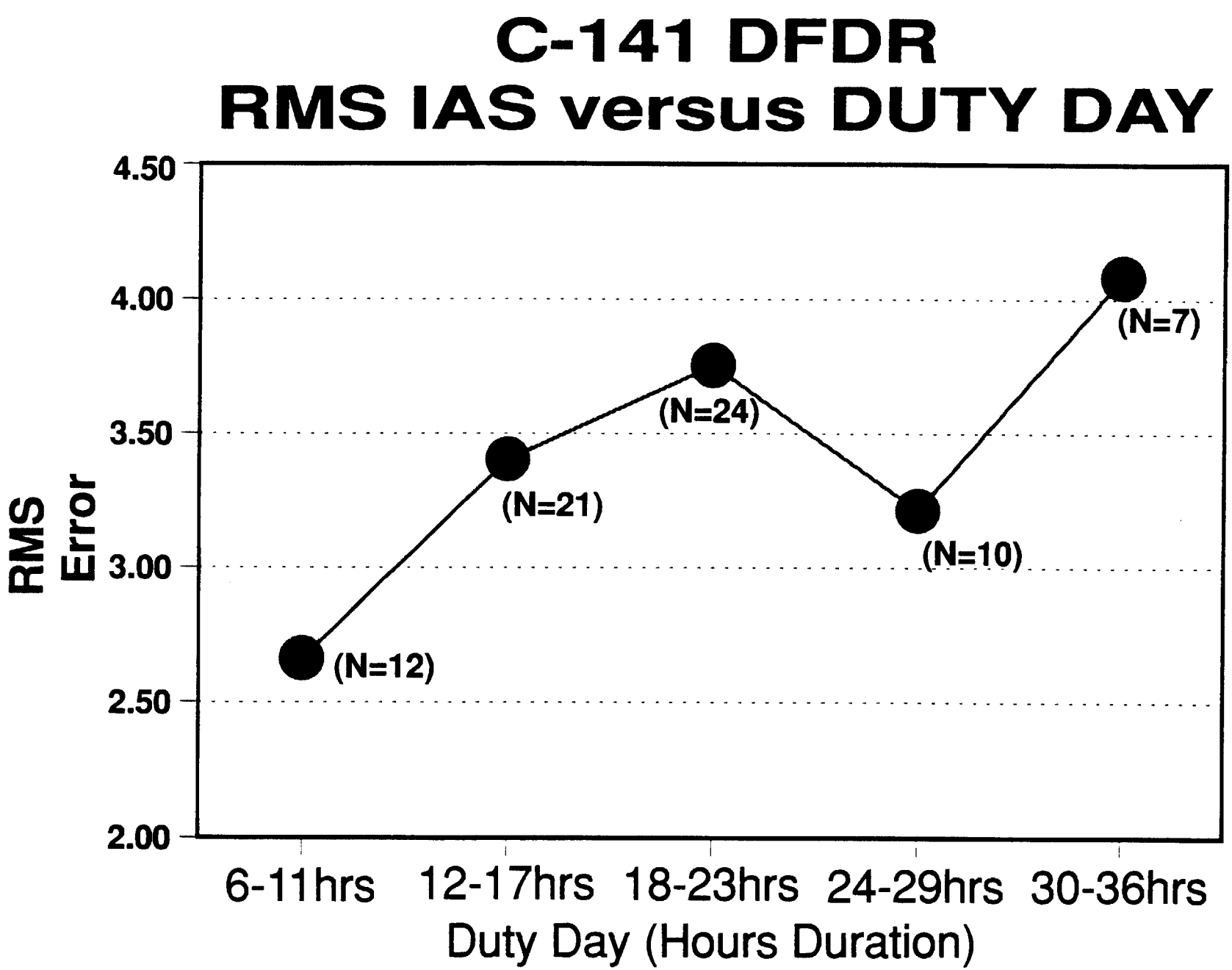
A	B	C	D	E	F	G	H	I	J	K	L	M	N
File vc620329.dat :: Landing 4 of 4 (ILS - EDAF) Night/VFR													
index	frame	time	alt	ias	head	pitch	roll	epr	gear	rev	spoil	flaps	hf1
483	6105	32	111	132.3	75.2	0.786	2.398	1.213	DN/LOCK	LOCKED	1.06	98.9	KEYED
483	6172	31	102	132.3	75.5	1.116	2.728	1.208	DN/LOCK		1.06	0.0	
483	6239	30	98	131.1	75.5	0.786	2.728	1.097	DN/LOCK		1.06	98.9	
483	6306	29	89	131.1	75.9	0.458	2.069	1.108	DN/LOCK		1.06	0.0	
483	6373	28	85	131.1	75.9	0.458	2.069	1.213	DN/LOCK	LOCKED	1.06	98.9	
483	6440	27	80	131.1	76.2	0.786	3.061	1.206	DN/LOCK		1.06	0.0	
483	6507	26	71	129.8	76.2	0.132	3.730	1.097	DN/LOCK		1.06	98.9	
483	6574	25	63	129.8	76.2	-0.518	4.067	1.108	DN/LOCK		1.06	0.0	
483	6641	24	58	129.8	76.6	-0.194	3.730	1.213	DN/LOCK	LOCKED	1.06	98.9	
483	6708	23	49	128.6	76.6	0.132	3.061	1.204	DN/LOCK		1.06	0.0	
483	6775	22	36	128.6	76.9	0.132	2.728	1.097	DN/LOCK		1.06	98.9	
483	6842	21	28	128.6	77.3	0.458	2.398	1.112	DN/LOCK		0.95	0.0	
483	6909	20	19	128.6	76.9	1.116	1.416	1.215	DN/LOCK	LOCKED	0.95	98.9	
483	6976	19	10	128.6	76.2	1.447	1.742	1.217	DN/LOCK		0.95	0.0	
483	7043	18	1	128.6	75.9	1.779	1.416	1.116	DN/LOCK		0.95	98.9	
483	7110	17	4990	127.3	75.5	2.112	1.742	1.138	DN/LOCK		0.95	0.0	
483	7177	16	4986	127.3	75.5	2.447	2.069	1.254	DN/LOCK	LOCKED	0.95	98.9	
483	7244	15	4981	127.3	75.2	2.447	1.092	1.258	DN/LOCK		0.95	0.0	
483	7311	14	4977	127.3	74.8	2.112	0.770	1.239	DN/LOCK		0.95	98.9	KEYED
483	7378	13	4973	126.1	74.8	1.447	0.770	1.269	DN/LOCK		0.95	0.0	
483	7445	12	4964	126.1	74.5	0.458	0.770	1.274	DN/LOCK	LOCKED	0.95	98.9	
483	7512	11	4959	126.1	74.5	0.132	1.416	1.256	DN/LOCK		0.95	0.0	
483	7579	10	4951	126.1	74.1	0.786	3.061	1.247	DN/LOCK		0.95	98.9	
483	7646	9	4951	124.4	74.1	8.904	2.012	1.101	DN/LOCK	EXTENDED	0.95	0.0	


Vertical Velocity Versus Time
EDAR (AC - Day/VFR)
Recent Sleep:
24hr = 10 hours
48hr = 12 hours
Duty Day: 28.0 hr
Moderately Tired; Let Down
Temp: 96.8
POMS: 43
Mean = 470 fpm
Vertical Velocity (fpm)
2000
1500
1000
500
0
-500
-1000
140
130
120
110
100
90
80
70
60
50
40
30
20
10
Time to Landing (Secs)


Vertical Velocity versus Time
EGUN (AC - Day VFR)
Duty Day 9.0 hrs
Moderately Tired; Let Down
Temp: 96.4
POMS = 41
Recent Sleep:
24hr = 4 hours
48hr = 14 hours
Mean = 575 fpm
Vertical Velocity (fpm)
2000
1500
1000
500
0
-500
-1000
140
130
120
110
100
90
80
70
60
50
40
30
20
10
Time to Landing (Secs)

C-141 DFDR
RMS IAS versus Subj Fatigue




C-141 DFDR
RMS IAS versus DUTY DAY
RMS Error
4.50
4.00
3.50
3.00
2.50
2.00
(N=12)
(N=21)
(N=24)
(N=10)
(N=7)
6-11hrs
12-17hrs
18-23hrs
24-29hrs
30-36hrs
Duty Day (Hours Duration)

Digital Flight Data
A Defense Conversion Opportunity

- Concerns
  - Cost
  - Validation
  - Operational Interface
  - Data security
- Operational Events
  - IAS, Pitch, Roll, VVI
  - Embedded Performance Tasks

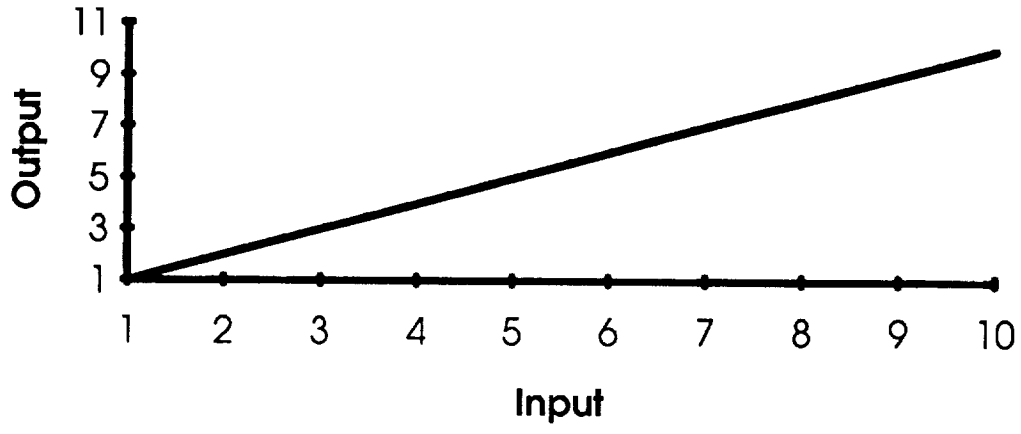
Digital Flight Data
A Defense Conversion Opportunity

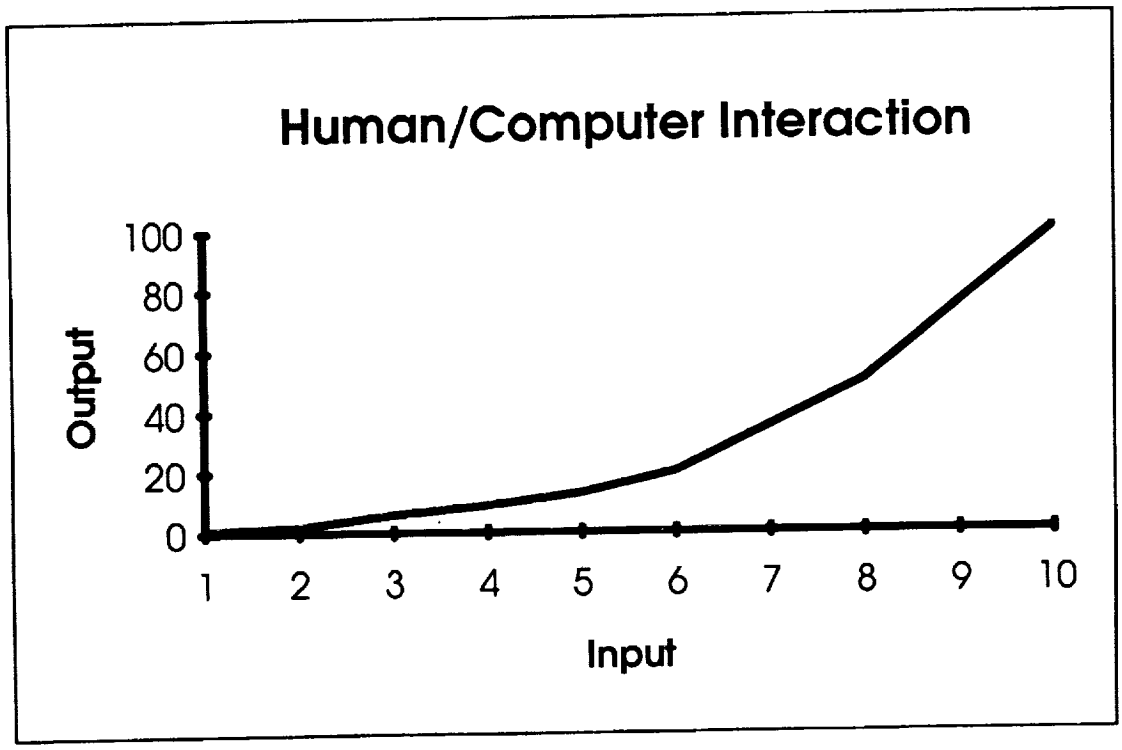
- Operational Requirements
  - Accident Investigation
  - Pilot/Crew Inflight Performance
  - Training
  - Flight Safety
  - Self-Assessment/Improvement
  - ATC Operations Improvement
  - Airport Departure/Approach Design
  - Aircraft Design

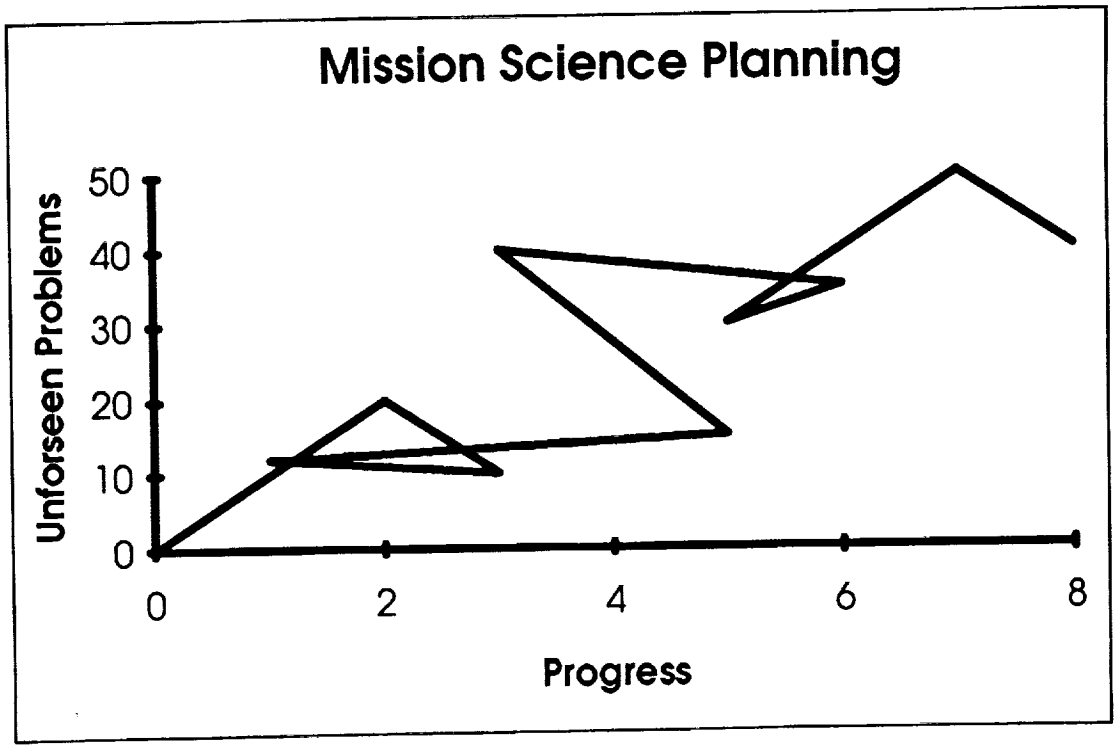
Science Operations
SLS-1 and SLS-2

Dr. Howard Schneider, NASA/JSC

Human/Simple Machine Interaction




Human/Computer Interaction
Output
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
Input


Mission Science Planning
Unforseen Problems
50
40
30
20
10
0
0
2
4
6
8
Progress

Life Sciences Goals

- Ensure the health, wellbeing, and productivity of humans in space
- Develop an understanding of the role of gravity on living systems
- Expand our understanding of the origin, evolution, and distribution of life in the universe
- Promote the application of life sciences research to promote the quality of life on Earth

SLS-1 and -2 Experiment Selection

Investigations were selected through a peer review process and were judged on scientific merit and relevance to the space program

- 25 Experiments were selected from approximately 400 proposals for Spacelab-4
- Spacelab-4 was divided into two missions, SLS-1 and SLS-2
  - SLS-1, an extremely successful mission, flew in June 1991
  - SLS-2 is scheduled for September 1993
    - Most of the experiments from SLS-1 will be flown on SLS-2 with some enhancements
    - SLS-2 will provide a larger sample population for the investigations and will verify SLS-1's research findings

SLS-1 Primary Payload

10 Investigations using humans

- 4 Cardiovascular/cardiopulmonary
- 2 Musculoskeletal
- 3 Regulatory Physiology (including human lymphocyte study)
- 1 Neuroscience

SLS-1 Primary Payload (continued)

7 Investigations using rodents

- 4 Musculoskeletal
- 2 Regulatory Physiology
- 1 Neuroscience

1 Investigation using jellyfish

- Neuroscience

Hardware verification

- Functional tests were performed on the Research Animal Holding Facility (RAHF) and the General Purpose Work Station (GPWS)

Constraints of Space Flight Research

Limited number of flights

Small subject population

- Shuttle crews for life sciences mission consist of 7 crew members
  - 3 crew members are responsible for Orbiter operations
  - 4 crew members are responsible for science operations

Constraints of Space Flight Research (continued)

Resource Constraints

- Crew time is limited
  - A typical day consists of about 6 hours for science payload activities
    - Operator duties and maintenance activities also occur during this period
- Experiments must function within spacecraft resources
- The spacecraft is a remote facility

Use of Animals

- Animals are being validated as models for humans
- A larger number of subjects can be obtained
  - 24 rodents can fly in 1 RAHF
  - 2478 jellyfish flew on SLS-1
- Invasive activities can be performed on animals

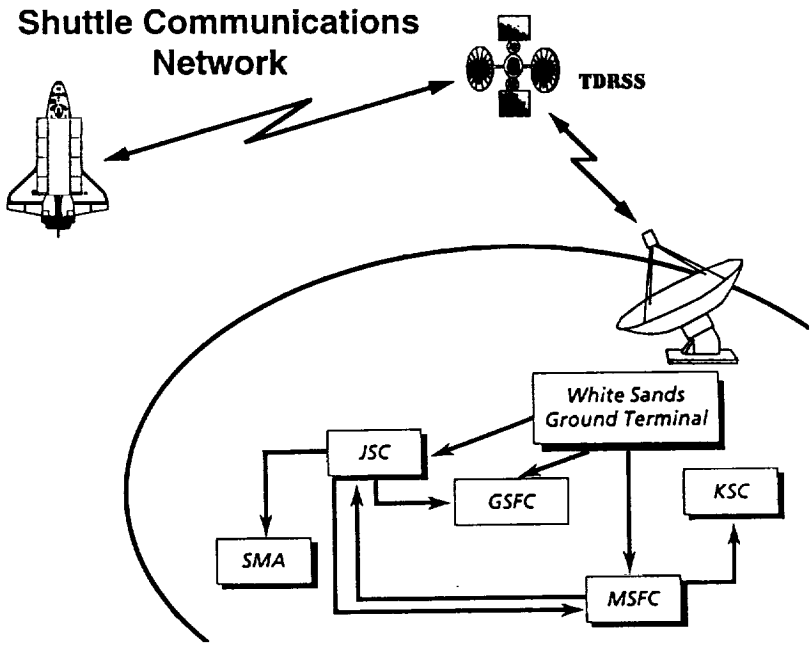
Scheduling

- Eliminate duplicate data collection through data sharing
- Share resources where possible
- Plan on activities to take longer in space (zero-g factors)
- Have a shopping list ready for each flight day

Communications

Communications with both the investigators and the crew during the mission is important

- Science Operations Planning Group (SOPG) meetings provided the investigators a forum to
  - Express issues and concerns
  - Report daily accomplishments
  - Review replanning activities
- Air-to-ground communications with the crew
  - Track crew status
  - Prepared to respond to anomalies



Hardware

Due to the closed environment of the spacelab, maintenance activities should be scheduled frequently during the mission for critical hardware

- Refrigerator/freezer filter servicing
- UMS servicing

Fly backups for critical hardware whenever possible

- Gas Analyzer Mass Spectrometer (GAMS)
- Urine Monitoring System (UMS)
- Echo

Contingency Planning

Plan for anything that can go wrong!

- Develop detailed contingency plans for precious samples
  - Document preflight where each sample is located
  - Track and update during the mission
- Establish priorities prior to the mission
- Launch delay timeline

Postflight Data Distribution and Reporting

Distribution of data postflight is an important element and a plan should be developed before flight

- Experiment data
- In-flight photography

A schedule for postflight reports should be established preflight

- 30 Day "Quick Look" report
- 180 Day Report
- 1 Year Report

Summary

- Integrate investigations to share hardware, protocols, and samples where possible
- Develop detailed contingency guidelines and procedures
- Establish priorities for all payload elements prior to flight
- Apply knowledge gained from each mission to future missions
- Establish a plan for postflight data distribution and reporting

Shuttle Operations: Recent Experience

John Muratore, NASA/JSC

Shuttle Operations: August 1993

The Space Shuttle is now the major part of the United States human spaceflight experience

57 missions over 12 years have accumulated over a full year of on-orbit experience

- Only 29 human piloted missions prior to the start of the Shuttle Program (Mercury, Gemini, Apollo, Skylab, and Apollo-Soyuz combined)

Using numbers of experiments or pounds of payload as a metric, the Space Shuttle is the most productive United States human spaceflight program

Shuttle Operations: August 1993

The Shuttle Program has demonstrated a surge capacity of one flight per month

Average flight rate of 7-8 flights/year has been conducted since 1990 while lowering the operations cost 5% per year

Shuttle Operations: Recent Experience Slide 2

What Is Mission Operations?

Mission Operations includes:

Facility Development and Maintenance – control centers, simulators, and off-line facilities

Flight Design – ascent, orbit and entry trajectory, consumables, Remote Manipulator System (RMS) Operations

Reconfiguration – Flight Software, Mission Control Center, and Shuttle Mission Simulator flight-specific software build and testing

Flight Planning and Procedures Development

Payload Integration

Crew and Flight Controller Training

Flight Operations – monitor and control of systems, trajectory, and payloads

Shuttle Operations: Recent Experience Slide 3

Mission Operations Costs: The Facts

Mission Operations is about 10-12% of Shuttle mission's cost (computed on a yearly basis)

Mission Operations has reduced costs by 15% since 1990 and is committed to reducing it another 15% in the next 2 years

- "Drain the Swamp" Total Quality Initiative

Most of mission operations resources (people and dollars) are spent in facilities development and maintenance, flight design, and reconfiguration

- Most of this cost involves software maintenance and operations

Actual flight operations (people on the consoles during missions) is a very small percentage of total mission operations

Shuttle Operations: Recent Experience Slide 4

Flight Operations Costs

Flight operations teams work the Mission Control Center 24 hours/day 7 days/week during a mission

Based on the flight phase, between 50-80 people/shift are responsible for monitor and control of Shuttle systems, payloads, trajectory, and flight planning

- This number significantly reduced since early Shuttle flights even though scope of on-orbit activities has increased

Based on the flight phase, another 80-100 people/shift are responsible for keeping all of the control centers systems operational (telemetry, command, voice, network, computing, display, electrical power, air conditioning, security, etc.)

5-6 teams are necessary to fly a rate of 8-12 flights/year

- 3 flights in 3 months in a surge requires 12 flights/year capacity

Further manpower reduction activities are under way relying on conventional and expert system based automation

Shuttle Operations: Recent Experience Slide 5

Shuttle Operations: August 1993

What is our experience level?

A test flight program of 57 flights would be considered short for a new commercial or military aircraft

- YF-22A Demonstration/Validation program had 74 flights with 91.6 hours
- X-29 accumulated over 200 flights in the basic flight test

Our ascent and entry experience base, although larger than any previous United States human spaceflight program, is still relatively small

Our on-orbit experience base is considerably larger with almost a year of on-orbit flight time

Shuttle Operations: Recent Experience Slide 6

Shuttle Operations: Ascent/Entry

Ascent/Entry environment remains very challenging and there still is a lot to learn

For example: Upper atmosphere density shear on STS-57 required 330 more pounds of Reaction Control System propellant than previous Three Sigma analysis experience

Margin has to be maintained in Shuttle operations and teams trained to deal with the unexpected

Activities to improve margin have been very successful

For example: Day of Launch I-Load Update (DOLILU) allows updates to flight software on the day of launch to account for variations in upper atmosphere winds. This capability has significantly decreased the number of scrubs due to upper atmosphere winds while increasing flight margins

We continue to fly conservative flight tests during operational flights to expand envelopes

Flight control teams and flight crews continue to be trained to deal with unexpected events and to provide "human" margin

Shuttle Operations: Recent Experience Slide 7

Orbit Experience

The Shuttle is relatively "forgiving" in the orbit environment

Shuttle systems perform very well with a minimal anomaly rate on orbit

- Testimony to the excellent work done at KSC in preparing the flights and to the care and expertise of the design community
- Recurrent problems (tv cameras, text and graphics system) are being addressed by upgrades

Shuttle Program has demonstrated the ability to integrate systems upgrades into the fleet without interrupting flight schedules

- New onboard computers, Mass Memory Units, Inertial Measurement Units, Star Trackers, TACANs, Fuel Cells, Auxiliary Power Units, Thermal Impulse Printer System (TAGS replacement), waste collection system have all been integrated into the fleet since STS-26 Return to Flight
- Next big activity is upgrading the Shuttle to "glass cockpit"

Most of orbit activity is dedicated to payload and experiment operations

- Which is the way it is supposed to be

Shuttle Operations: Recent Experience Slide 8

Shuttle Payload Integration

Shuttle places relatively few constraints on payload operations (safety, etc.)

- Shuttle provides a good microgravity environment
- Shuttle has shown itself to be an excellent attitude and pointing platform for earth pointing, sky pointing, or co-orbital pointing payloads

- Constraints to payload operations still exist however with most payload integration issues involving interfaces BETWEEN multiple payloads, not issues between the payload and the Shuttle
- Typical integration issues involve:
  - Attitude and pointing between multiple payloads (including thermal issues)
  - Mission duration, peak power, power outlet allocation
  - Stowage
  - Forward RCS fuel can be limiting in multiple rendezvous missions
  - Crew time allocation and support

Shuttle Operations: Recent Experience Slide 9

Orbital Activities

In-flight Maintenance is not a big driver on Shuttle missions

- Most IFMs are to maintain mission success
- Most of the IFMs are payload related, usually secondary or small payload related

Tracking and Data Relay Satellite (TDRS) Network is reliable and has made capture of huge amounts of scientific data possible

- Network is very busy and, although Shuttle has priority, we are constantly adjusting our TDRS usage to give other users a chance

Remote Manipulator System (RMS) is a mature system with tremendous capabilities

- Space Vision System demonstrated as a flight experiment on STS-52 has shown itself to be the next big improvement in robotics capability

Shuttle Operations: Recent Experience Slide 10

Orbital Activities (cont'd)

Extravehicular Activities (EVAs) development continues

- Significant improvements in the EVA work-system and EVA training have been made in the last year
- Regular program of EVA Flight Tests has been instated to rapidly develop techniques and capabilities

Shuttle Operations: Recent Experience Slide 11

Personal Computers

Personal Computers are becoming a major on-orbit tool

- Utilized for off-line computing for Shuttle use
- Utilized for real-time computing with real-time interfaces to payload and Development Test Objective hardware
  - Laser ranging devices primary crew use through the Payload General Support Computer (PGSC)
  - Major payload command and monitoring role in the STS-46 Tethersat mission
  - STS-51 will remonstrate real-time monitoring of Shuttle telemetry by onboard personal computing
- Linked to ground computers via modem interface over air-to-ground voice loops
- Onboard printer now also manifest

Contribution of Space Technology to Improved Aerial Reconnaissance Operations

Major Mark Pestana
Space and Missile Systems Center
Operating Location AW
NASA/Johnson Space Center

Contribution of Space Technology to Aerial Reconnaissance Operations

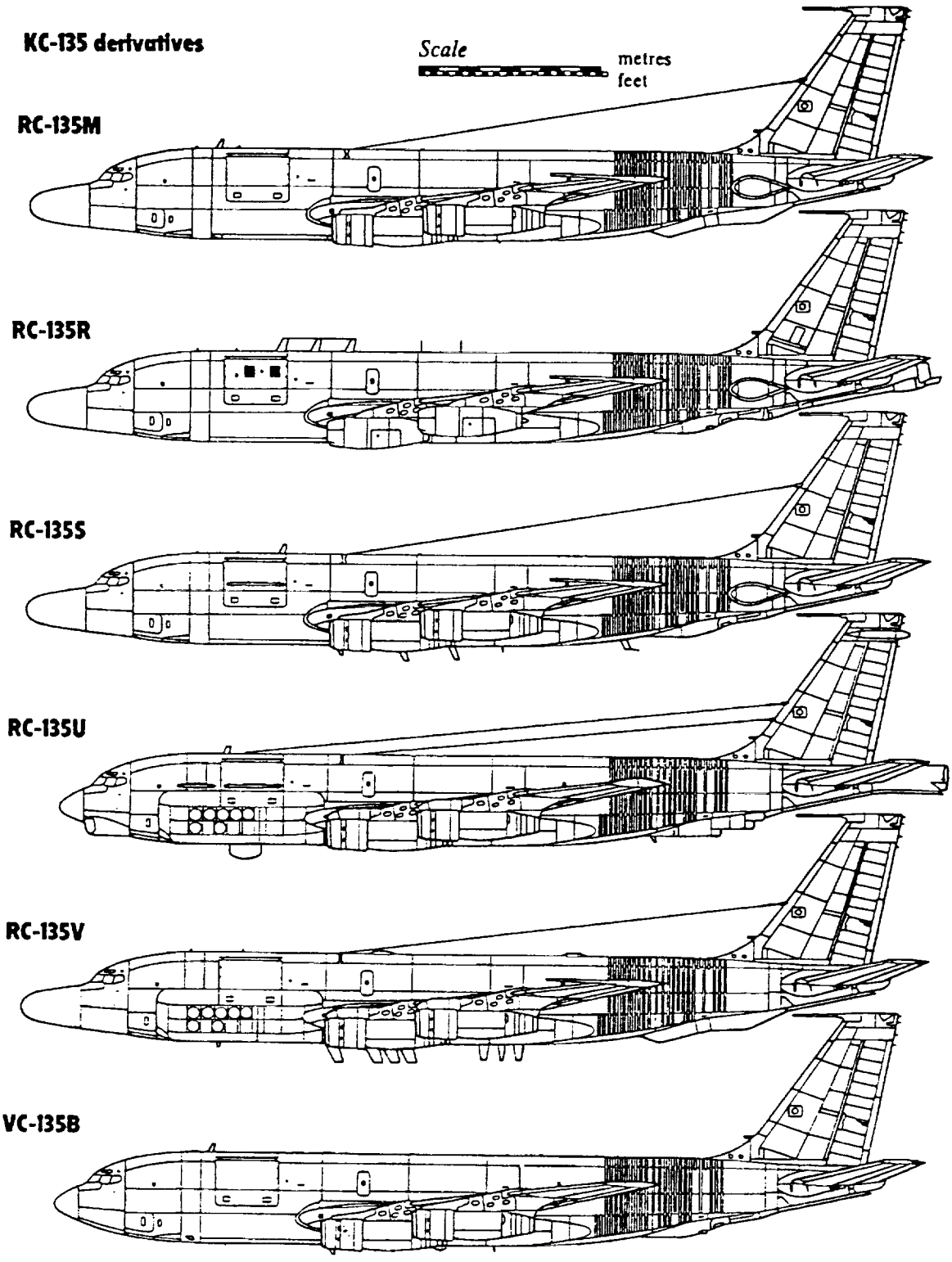
- RC-135 Aircraft Description
- Reconnaissance Mission Scenario
- Navigation Requirements
- Mission Success Dependencies
- Systems Limitations
- Space Applications Improvements
- Results

RC-135 Aircraft

- Boeing 707-type (KC-135: Boeing Model 717)
- Highly Modified for Electronic Intelligence
- 55th Recon. Wing. Offutt AFB, NE
  - 4 Permanent Overseas Operating Locations
- 33 - 35 Flight Crew Members
  - Pilot, Copilot, 2 Navigators
  - Electronic Warfare Officers
  - Linguists
  - In-flight Maintenance Technicians
- Long-duration Missions
  - 8 Hours w/o Aerial Refueling
  - 13 Hours with Aerial Refueling (Typical)
  - Extended Durations with Augmented Flight Crew

Recon Mission Scenario

- 13-hour Mission Duration
  - Approximately 50%-70% within "Sensitive Geopolitical Area"
  - Over International Waters or "friendly" territory
- Rendezvous with Tanker
  - Outside the Sensitive Area
- Gather Electronic Intelligence (ELINT)
  - SIGINT
  - COMMINT
- Capability for Increased 24-hour Tasking
  - 2 to 3 aircraft on continuous regeneration cycle


KC-135 derivatives
Scale
metres
feet
RC-135M
RC-135R
RC-135S
RC-135U
RC-135V
VC-135B

Navigation Requirements

- Rendezvous with Tanker
- Two Navigators operate separate equipment
  - NAV 1: Stellar Inertial Doppler System (SIDS)
    - INS with Star Tracker
    - Doppler Radar determines ground speed
    - SIDS can fly predetermined track through Autopilot
  - NAV 2: Radar and Sextant
    - Position fix on land masses
    - Sun or star fix
- Crosscheck positions at specific intervals
  - Discrepancy in positions may require mission abort
- NAV systems also key to SIGINT collection
  - Precisely locating intelligence targets depending on NAV

Mission Success Dependencies

- Command, Control, Communications
  - VHF and UHF
    - Air Traffic Control, Tanker Coordinator
  - HF
    - Worldwide Command and Control
    - Mission monitoring, status reporting
- Navigation Accuracy
  - Precise Positioning is Essential and Critical
    - Cannot violate sovereign airspace
    - Determination of target locations
- Threat Assessment and Timely Warning
  - Internally via specialists on board
  - Externally via HF

HF Communication Limitations

- Subject to Solar Interferences
  - Increased solar activity affects ionosphere
- Used to transmit data intermittently
  - Awkward coordination of radio usage between users
- Voice transmission is not secure
  - Must take time to encode and decode messages
- High static noise levels
  - Fatigue factor for long-duration missions

Navigation Limitations

- INS position will "drift"
  - Periodic updates required
    - Automatically via Star Tracker
    - Manually via NAV 2 systems
- Polar regions
  - Cannot rely on position updates from radar
- Weather conditions
  - High overcast blocks stellar positioning
  - Severe weather can degrade radar positioning

Mission Improvements Via Use of Space Technology

- Air Force Satellite Communication (AFSATCOM)
  - Terminal in cockpit (keypad and printer at NAV 2 position)
  - Multiple comm links available
  - Can build and store standard messages
  - Secure: encrypted automatically
- NAVSTAR Global Positioning System (GPS)
  - Multiple-satellite constellation: always available
  - High accuracy
  - Differential GPS applications
- Defense Meteorological Satellite Program (DMSP)
  - More accurate forecasting in remote areas

Results of Incorporating Space Applications to Existing Operations

- Operational Effectiveness Improved
  - Accurate target location
  - Overall intelligence "picture" is enhanced
- Operational Efficiency Improved
  - Less dependence on "single-point failure" systems
  - Timely, secure communications
  - Tanker support optimized
- Safety Improved
  - Weather forecasting of route and destination
  - Accurate and timely threat assessment

PANEL DISCUSSION HIGHLIGHTS

PANEL DISCUSSION HIGHLIGHTS

OPERATIONS CHALLENGES

Moderator:
Dr. Kumar Krishen, NASA JSC

Panelists:
Dr. Melvin Montemerlo, NASA HQ
Mr. Gael Squibb, NASA JPL
Mr. John Muratore, NASA JSC
Maj. Kory Cornum, USAF

KRISHEN: Welcome to the panel discussion. Our theme is operational challenges. We have actually five panelists, because I'm going to include myself as a panelist.

Our first panelist, Dr. Mel Montemerlo, is manager of the artificial intelligence research and development program in NASA's Office of Advanced Concepts and Technology. Mel has been in that position since the program was initiated in 1985 as a result of Congressional interest in automation and robotics technology for Space Station, for NASA in general, and for the spinoff to U.S. industry. In recent years, Code C has been charged as the organization that reflects this particular technology program in their broader discipline area – and the broader discipline area is called operations. Mel got his B.S. in mathematics from Catholic University in 1964, his M.S. in math from the University of Connecticut in 1966, and his Ph.D. from Penn State in 1969. He has spent a lot of his career in the automation and robotics field, and it will be interesting to see his points of view in this panel.

Then we have Maj. Kory Cornum. He's a graduate of the U.S. Air Force Academy and the Uniformed Services University of the Health Sciences School of Medicine. After attending both U.S. Army and U.S. Air Force flight surgeons classes, Maj. Cornum served with the 20th Special Operations Squadron for 1 year. He then began serving with the 58th Fighter Squadron at Eglin Air Force Base in 1988. Maj. Cornum was deployed with his squadron to Saudi Arabia in support of operations Desert Shield and Desert Storm in August 1990. This fall, Maj. Cornum will be moving to the Armstrong Laboratory at Brooks Air Force Base.

Mr. John Muratore is the Flight Director in our Operations Directorate at Johnson Space Center – prior to that he was the Chief of the Flight Software Reconfiguration Division at JSC. He has been pioneering the use of artificial intelligence and expert systems in our Mission Control Center here.

KRISHEN: (Cont'd)

Mr. Gael Squibb has a depth of knowledge and experience in operations – both in the development side as well as the operations end of operations technology and research. He was with Lockheed Missiles and Space from 1962 to 1964 where he worked in operations at the U.S. Air Force Satellite Control Center. He joined the Jet Propulsion Laboratory and held positions as Assistant Flight Director for Soviet Missions, Project Manager Infrared Astronon Nickel Satellite, Infrared Processing and Analysis Center Manager. In the development arena, he was detailed to ESA from 1988 through 1990, where he worked on the requirements and functional design of Infrared Space Observatory Science Operations Center. He joined the Smithsonian Astrophysical Observatory in Cambridge, Massachusetts, in 1991 and managed the AXAF Science Center. Then he returned to JPL in March 1993, where he's the Manager of the Mission Operations Development Office.

What I'd like to do is give you a brief synopsis of our committee – who we are, the STIG Operations Committee – and then I want to charge all of us and challenge all of us, to make this a very productive panel discussion. We are getting a video tape of this particular session, and we'll share this video tape with many other members of our team who could not be here and with those who are interested in this operations world.

The Space Technology Interdependency Group has eight committees. At NASA Headquarters, Mr. Greg Reck is the cochairman of this group; and from the U.S. Air Force side it's General Paul. The steering committee generally consists of members from Headquarters, and we have in the steering committee U.S. Navy involvement, the U.S. Army, Department of Energy, ARPA, and several other organizations. As you see, our theme is actually operations. The word *Space* should come out of it. We are responsible for both ground and space operations. So we call ourselves the STIG Operations Committee, and SOAR is one of the activities we sponsor.

This committee came up with a brainstorming session about a year or two ago, and we have been refining some of these goals. Now we're going one step beyond this. We're trying to measure our performance, which is probably going to be even harder than setting the goals and doing what we want to do.

The goals are: Identify and characterize interdependent activities, activities that could be done together by several organizations – initially we started with Government organizations and now are quickly expanding it to academia and industry. The second goal is to encourage interdependent programs. The third one is to interchange technical and programmatic information and share lessons

KRISHEN: (Cont'd)

learned. We had a fantastic plenary session this morning, talking about operations experiences and the programmatic information we share in terms of two meetings that we have on a yearly basis. We meet twice as a committee to identify political works and nonproductive overlaps. That's a challenge to us; that's a challenge to our committee. In the sense that, sometimes it doesn't matter – even if four or five agencies or industries are doing robotics and automation work – that doesn't mean that all the important technologies and research are being addressed. One of our challenges here in this symposium, and as a committee, is to find those works that are left out. Finally, in the last year it dawned on us one of our critical themes also should be to promote technology to industry and academic institutions. What we learn in terms of operations experience should find its way into industry and academic institutions – and vice versa, too.

We sponsor the Space Operations, Applications and Research Symposium and Exhibition and we have goals regarding that. One of our challenges in the operations committee was to get organized, and to show that we are an efficient committee, an efficient team, in a cost-effective manner. That's why we have come here with a minimum menu. We have one symposium exhibition in a year, and we have only two meetings where we meet and the rest of the correspondence and the rest of the communication we do are via telephone and faxes.

We organized ourselves along the lines of five disciplines that were okayed by our Headquarters STIG group. These are: robotics and telepresence, automation and intelligent systems, human factors, life sciences, space maintenance and servicing. Some of these are real productive research and technology areas, as you well can imagine.

All the members reflected here (I'll show you five more vugraphs), all of us are equal members in this committee. As you can see, we have pretty good participation from not only NASA and the U.S. Air Force, but U.S. Army, U.S. Navy, and the Department of Energy. We feel lucky that our committee has such a wide variety of membership.

We conduct two meetings and facilitate the communication of our R&T results in operations and across the Agencies. We want to document the results of our R&T in the operations world and one of our means is the symposium proceedings that come out of this particular symposium. I said earlier we are reflecting both ground and space operations. Provide interface with our committees: That's also one of the jobs we have to do, and we have not necessarily done the best in that area. We need to loop with those seven other committees much better.

KRISHEN: (Cont'd)

Now I'll show you very quickly the five committees we have, so you can get an appreciation of the charter that those committees held. This is the robotics and telepresence subcommittee. All the telepresence, telerobotics, robotics, dexterous manipulation, space maintenance and assembly having to do with robotics and automation, all of these come under their prerogative.

This is the automation and intelligent system subcommittee. Knowledge-base systems, knowledge capture, expert systems, artificial intelligence, neural networks are some of the things that are right now "growing things." Vehicle health monitoring is very important.

This is the human factors subcommittee. There's a lot of activity in this subcommittee in regards to human performance measurement and prediction, extra- and intravehicular operations, human-machine interface, training systems, virtual reality is getting to be a hot subject, performance characterization, crew training systems.

We get approval from the Headquarters STIG group to put members on these subcommittees or the whole committee, but we're always open to new members. In the future, we'll get probably a little more instruction to get membership from academia and industry.

The life sciences subcommittee has been with us all these years. It deals with life support, health systems, biomedical research, medical operations, space radiation effects. There is a committee under STIG that has to do with radiation in general, but we are concerned here with radiation as it pertains to life sciences – primarily that part of it.

This is the space maintenance and servicing committee. It covers the scope of maintenance and repair operations, assembly operations, servicing operations, and this morning's presentations were really interesting in my viewpoint on some of those issues. How to grab Intelsat, for example. Hull detection and nondestructive evaluation.

Last year, we felt as a committee that we needed people who had lots of experience to tell us something about operations experiences in terms of what they expect to plan, what they didn't expect, and what happened. You saw some examples of that this morning. Those were the operations experiences, because operations experiences to us means that we give them the systems, the software, and all the technology and the product of our research, and whether it's an astronaut or a flight controller or a flight director, whatever, they try to use that R&T

KRISHEN: (Cont'd)

and experience something and then try to relate that experience – good or bad. Then we said we must also deal with operational challenges. Challenges that I see personally are in terms of: How do you reduce the cost? How do you increase the safety? Tell me how you can incorporate some new capability that we haven't seen before? How to grab a satellite. You had to produce that arm and, in this case, we couldn't grab with the arm. Then we had to produce the three astronauts to grab the Intelsat. But the point is, tell me how this new capability will work. I was asking Dr. Schneider, "Have you thought of using MRI (magnetic resonance imaging) in space to try to understand the real time; for example, what happens to a bone or a muscle and things like that?" And, his answer was very appropriate. He said, "We can't even take one into space." That's a challenge for us; that's a technological change. How do you give them the capability to monitor the operations based on new technologies?

Then the last item: What we meant by environment, I want to be very careful to tell you. What we're trying to say is, the challenge is to keep the environment the best we know. If we have altered the environment in a bad way, we must get it back to a better state. That's what I meant by operational challenges. We will dwell in this particular symposium on these two issues, and out of them will come some research and technology. And, that's the challenge we all have here – all of us who are attending this symposium – is to somehow identify some new research and technology thrusts and then go ahead and develop the mechanism. That means we have to convince our bosses and whatever agencies to develop those technologies.

Finally then comes this evaluation loop. For example, people like Muratore are getting involved in this trying to tell us, How good did we do? Once this technology is incorporated, implemented, what is its performance? Basically all this translates into what I call the technology transfer process. So the panelists this afternoon have the challenge to communicate to you various aspects of these "operational challenges." Then all of us as the audience have the challenge to come back and try to figure out ways and means of alleviating those concerns that we come up with. Doing research and technology that will somehow give the performance that we're looking at.

That's what the theme of this particular symposium is, we're talking about operations experiences. We're going to talk about operations challenges. We're going to listen to some people talk about their research and technology results. And, we are also going to get some ideas about how these R&T thrusts are being implemented and what their performance is.

KRISHEN: (Concl'd) This whole thing is a technology transfer process. To that, the last word I have to say has to add another dimension: which is, now we're supposed to also think about how to transfer this technology to the commercial world. With that, I'm going to ask Dr. Montemerlo to go ahead and give his presentation.

MONTEMERLO: Thank you very much, Kumar. I guess it's a hard thing that Kumar just asked us to do. The way I think about it is by thinking of a person named Hammurabi. You all remember Hammurabi? Hammurabi came up with the first codified set of laws. But, he also did something else a lot of you may not know about. He invented apple pie.

Now codified laws have stuck around, but apple pie didn't. Apple pie really didn't show up again until the late 1800s when it was reinvented by Mrs. Smith. There are a lot of similarities between the way Hammurabi did it and the way Mrs. Smith did it, but there is a major difference which we have to keep in mind today. The reason apple pie died was the way Hammurabi cut it. You see, he had it the same way Mrs. Smith did and the way we still have it now, in that there's a soggy crust on the bottom, apples in the middle, and a crispy crust on top. He also did six or eight slices, but they were all parallel to the table. So the first slice was all crispy crust, you know the next six were just apples with a thin ring, and the last person got the soggy crust on the bottom. Mrs. Smith did the same kind of pie; she just cut it differently and it was a lot more useful, and the idea stuck around.

So the important thing for us is to come up with ways of cutting this operations pie in such a way that we can explain to those we need to explain it to why it's important that we work it. Why we develop new technology. And, how we get it implemented.

The way I see it at NASA is, currently the requirements for operations, we have Shuttle, Hubble, TDRSS, DSN, all sorts of things. Coming soon, hopefully, a Space Station, EOS, AXAF; after that a lot more things. Meanwhile in terms of operations, let's not forget we have a large NASA infrastructure. We have procurement; we have insurance; we have travel; we have an awful lot of things which I believe we could cut back. With existing technology, I believe we cut back the cost of doing procurement and travel 20% in nothing flat. We ought to think about that. Sometimes we don't have that in our work breakdown structure. We need to put it in.

My operations program used to be called artificial intelligence. They changed the name to operations. The problem with the term artificial intelligence is that it didn't cover everything we did. However, the term operations describes far

MONTEMERLO: (Cont'd)

more than we ever can do. So this problem of words is a little difficult, but the push now is to describe things – technologies – as to where you apply them so your customers can understand where they should come to. So that's the one of the new ways of thinking back in Code C.

Our goals are to reduce mission life cycle costs. I'll say something about that at the next slide. But, life cycle is important. Reduce the marching army with no decrease in capability, and there's an awful lot we can do there. More bang for the buck in science. Oftentimes in science where you have one PI or a few number of scientists we can't analyze all of the data we have. So if we can analyze more of the data in more interesting ways, there's an awful lot we can do. I consider that part of operations. We need the product ties, those operations technologies, and we need to get them out to industry so we can make future missions affordable.

I've tried to get a handle on how big operations is in NASA. It's hard to write that down, to find it anywhere. Best I can tell, Shuttle operations is somewhere over $3B a year, maybe $3.5B a year. The number of people who touch Shuttle in an operational way, who are paid for by the operations cost of Shuttle, inside and outside of NASA, from what I understand, is around 25,000 people. That's a lot of people. I think we probably could do that for less now, and we certainly can't use that as a paradigm for Station.

If you look at science, somewhere between $500M and $1B are spent for science operations, depending on how you count. If we're a $14B a year organization, we may spend $4.5B on operations. We have a lot of things we can do to improve that.

In terms of the operations cost, a lot of the problems we have later on in the design of something – or in the operation of something – is the fact that we didn't capture the knowledge during the design phase. We need to do better on integration and test what we commonly know of as operations and also science. That's how I cut the world.

NASA operations challenges. This was a challenge to write down on a slide. And, this is my one list of challenges. I'm not proud of it. I didn't generate these; I collected them. The way I look at it is, we can look at manned operations, science operations, and I think of those first two as a lot of what happens in Mission Control, mission operations. And later, there's data visualization and analysis. There are a lot of things we can do for autonomous spacecraft and then there's the infrastructure under crewed systems operations.

MONTEMERLO: (Cont'd)

Control room automation is something we've been working on for a long time. We've got a long way to go. But, besides what happens in Mission Control, there's scheduling, there's software engineering, there's training, there's electronic documentation. We're working on all of these, and there's more to do.

Another thing is, we have principal investigators all over the United States – all over the world in some cases. We want them to be able to operate from their home bases and to be able to work together; distributed principal investigators. Data visualization. We have roomfuls of data that haven't been analyzed. We need tools to easily analyze the terabyted data rates we're going to get from EOS. The Hubble images, which is another challenge. How do we analyze image data? We're doing a lot of work on this, but there's more to be done.

I pulled out virtual reality and other tools for data visualization. Virtual reality isn't the only way. Putting data in, in a way which you can see it and remember it, is very important. The whole visualization. How do you understand what's there. How do we do data snooping after we've looked for things we think are there.

We also need tools for engineering data analysis. Now, we need more autonomous spacecraft. If you look at the amount of work that goes into controlling uncrewed spacecraft at Goddard and JPL, it's amazing. We need tools for intelligent data compression. We need design techniques for intelligent vehicle health maintenance – for there as well as for crewed spacecraft. Intelligent instruments so that, when necessary, we send down information rather than data – send down crunched data. If we have small spacecraft, we're going to have lesser power sources, lesser bandwidth in our communications, and so we may have to go with LOSy data rather than LOSless. It would be better if we could send down information rather than data.

Under infrastructure: I did something a couple of weeks ago. I went to AAAI – American Association for Artificial Intelligence – which was held in Washington, D.C., which is where NASA Headquarters is. I tried to fill out the forms to get the registration fee and to do everything. Turns out that we have a new secretary who didn't know all the right ways to do it. Some of the other ones who had been around didn't know how to do this because, if you go to another city to go to a symposium, the fee for registration comes out of travel. But, if you do it from within D.C., then it comes out of training. There are all sorts of rules which change every so often, and I spent close to 3 hours trying to figure out how to get that done one afternoon.

MONTEMERLO: (Cont'd) I believe we could, with today's technology, put together smart forms which would allow people just to get this thing done on their Macs and PCs, right there at their desk. We could save an immense amount by using smart forms for procurement, payroll, personnel, logistics, insurance, travel. We need tools to capture and utilize corporate memory.

That's the way I see the overall set of things on one page. Now I didn't mean for anybody to read this. I wanted to make a point. The point is that, as you go across the top, you see words like TRANSPORTATION, STATION, ASTROPHYSICS, PLANETARY, SPACE PHYSICS, LIFE AND MICROGRAVITY, EARTH SCIENCES, COMMUNICATIONS, INFRASTRUCTURE. Those are the users of the technologies, many of which we developed in the AI program. If you look down another side, it's another way of cutting it: CONTROL ROOM AUTOMATION, LAUNCH PROCESSING OPERATIONS, MISSION PLANNING AND SCHEDULING, SOFTWARE ENGINEERING, TRAINING, DATA ANALYSIS. If you look, now you go across, what you see is a lot of the technologies we've developed are multiply applicable. And, that's good.

I think much of what we do, say in automated scheduling, is as applicable to Space Station as to Shuttle as to spacecraft as to making potato chips. It's easy for us to take and demonstrate something in one place and see its applicability in other places. Also, I think it's easy for us in this area of intelligent software to help in getting things commercialized. Because, by nature, much of work is multiply applicable.

The problem unfortunately is sometimes potential users, unless we demonstrate for them, don't see the applicability. For instance in PLANNING AND SCHEDULING, we haven't got enough money to do demonstrations for TRANSPORTATION and STATION and ASTROPHYSICS and LIFE AND MICROGRAVITY and EARTH SCIENCES and on and on. So we have to find some ways of getting our users together and have them see the applicability of some of these. That's a difficult challenge.

The meta challenges. One of the problems is, even if we have technology we can't always get that technology implemented. I can give you an example: When John Muratore originally tried to get his INCO ideas applied, he had difficulty with his management. He came to us, and we got him the money. Through a good deal of ingenuity on his part, he was able to demonstrate from within what you could get out of this and changed the minds of the people here, who could make the decisions as to what needed to be done.

I'll give you another example: Kennedy. We worked at Kennedy to try and improve the scheduling that's done for the processing of Shuttle. Well, the

MONTEMERLO: (Cont'd)

scheduling and much of the processing is done by support service contractors. One company is paid to do the scheduling by them. That company was also paid to put together an automatic scheduler. How do we entice the company to really put together a good scheduler to cut back on the cost of doing his scheduling? Just because we have a technology doesn't mean we can get it implemented.

There are a number of meta challenges to getting this implemented. One of them is changing the reward structure to facilitate cost cutting; to facilitate the insertion of advanced technologies into projects. I believe we need to insert into our reward structure – in contracts, in grants, and in the way we work with internal people at NASA – the incentives to do cost cutting and do change. Unless we do that, I think it's going to be a tough road. It's the old thing of rewarding managers to cut cost and staff.

Commercialization of dual-use technology. There are contractual impediments to getting things commercialized. If you try to do commercialization contracts, you know about some of those. But, we don't have to engineer the solutions right now. I'm just trying to lay out a list of the categories, a taxonomy, of what the challenges are and what some of the meta challenges are. I guarantee you that there are contractual impediments to commercialization.

We need to change the ways of doing business in research and development. I use John Muratore a lot because he's not only been successful but he's written about it, and he talks about technology transfer being a contact support and about technology transfer in tennis shoes. There's no doubt about it. We've got to get people who have the technologies together with the people who can use them. They have to live together; and, to get it done, we have to find ways of rewarding the researchers. If a researcher only gets rewarded by how many publications they get in jury journals, that's not going to cut it for us. We need other ways of rewarding them.

What I mentioned on the last slide was to help potential users see the multi-applicability of technologies, because we haven't got the money and the time, or the people, to do the same demonstration for each of the users. So what I see in NASA is a wondrous place to develop this technology and to get it applied. Most of the people I know in AI in NASA and around NASA came to work with us because we have the great and grand good things to do. They're space techies. They want to have an effect.

MONTEMERLO: (Concl'd) So the technology, some of it, is here. More of it is coming. We've got great places to apply it. I'm looking forward to it. I think we're going to do well if we find ways to take care of some of the meta challenges.

MURATORE: I want to talk to you a little bit about the challenges facing us in human space flight operations right now. The first and the biggest one, as I talked about this morning, we've been involved in since 1990. We've reduced the cost of mission operations at the Johnson Space Center by 15%, and we've got a goal of reducing it another 15% over the next 2 years.

We got hit with some pretty hard challenges. [But] when we looked at the problem carefully what we found was, there are ways to not only maintain safety and quality while reducing costs but actually to improve the safety and quality of how we do business while reducing cost. That's the big challenge out in front of us. Not to take more risks with the flight vehicle, but to find ways to do the job we're doing, improving the safety and quality while reducing the costs.

The second of which that's a big challenge is maintaining our knowledge and experience base. I was in this assembly the other day and I don't think of myself as a particularly graybeard NASA kind of guy. I turned to the flight controller, and I said, "Didn't that happen on STS-2?" this problem they threw in the simulation. The controller turned back to me and said, "Gee, Flight, I don't know. I was in high school then." And I went, shock!

We're having a big change in our knowledge and experience base. You know in the early programs, the people who were with us in Mercury and Gemini stuck through to the end of the Apollo. They stuck through Skylab. They came into the Agency young at the start of the program, and we've still got a bunch of them working with us today. That's changing.

I don't see that the problem of the knowledge base changing is going to be something that we can't move through and, in fact, fly well through. The problem is going to be, how do we capture the knowledge and personnel and procedures and training and software? Then how do we access that knowledge at the critical times that we need it?

We had an incident last flight where we sent a command to the vehicle that had some unexpected responses. The information to know the right sequence of commands to send was in the program. It was even available to the people who were sending the commands, but they were unable to access it properly. That's going

MONTEMERLO: (Cont'd)

to be a real problem. Not just having all the data and having all the knowledge, but being able to access it rapidly.

From an operations viewpoint with the Shuttle Program, a lot of what we're doing revolves around EVA. We're going to need a better EVA training process. The current water tank system has some strong limitations, and we've got some – as Rick Hieb told you this morning – pretty strong lessons learned. Maybe lessons relearned on Intelsat, based on how the water tank can fake us out. We have to understand the limitations of that process. I personally think we're going to need to try different training environments to make that work out.

Then we've got aging systems everywhere. We've got a large installed software and hardware base. We've got lots of aging systems supporting operations. I thought it was a unique JSC problem, and then I went around to the other Centers. At Goddard, they've got the same problem, at JPL, and at Kennedy. The systems we use to support our missions are all old. They're what are called legacy systems, systems that are so old that anybody's afraid to touch them or upgrade them anymore. So we've got to find a way of dealing with legacy systems.

Then the last one is improving the science return from the missions. Now what we've got fortunately is a whole bunch of technologies sitting out there that we think we're spending a lot of time and effort investigating. I think they offer rich fields for technologists to work with us in. One of them is global positioning technology. It offers the possibility or the capability of spacecraft that will autonomously navigate and significantly reduce our ground tracking requirements. On the flights STS-61 and STS-59, we're going to fly a GPS receiver and be taking a lot of data during all phases of flight on the effectiveness of the GPS system.

A hot buzz word: virtual reality. Virtual reality, I think, is where AI was in operations in 1986 or so. It was the hot buzz word then. It was the easy answer to lots of problems. And, it was just coming into the range where people could really use it for something. It's clearly got potential for crew training and EVA, for remote operation of spacecraft, and for training our flight controllers.

During the Hubble repair mission, which is this December, we're using virtual reality in two very different ways: We, the Engineering Director under Charlie Gott and Dave Homan, are first working with the flight crew. One of our problems in preparing for this mission has been integrating the activities of the people operating the RMS – the people inside the Orbiter – and the EVA personnel.

MONTEMERLO: (Cont'd)

On the Intelsat rescue, one of the strongest things we learned was coordination between IVA and EVA crewmen was a critical aspect. So the Engineering Directorate here at JSC – Dave Homan's and Charlie Gott's people, the IGOL lab – they're off working with the STS-61 flight crew, and the STS-61 flight crew has been over there regularly doing training.

Chris Culbert's group in ISD and the flight control team have been working on using virtual reality in a different way. Our concern is that there are a large number of people who are involved with planning and executing a Shuttle mission. The crew gets to spend a lot of time working with the actual flight hardware and look at high-fidelity mockups. Generally, the crew have a very good take on the exact hardware configuration. We send them up into space probably the smartest people on the problem, and they go up and they run into a problem. They call back down to the ground and say, "Hey, guys, go work this and, in the morning, brief us on it."

The problem is that we in Mission Control have limited training in this activity because we don't have as much access to the equipment and the simulators as we'd like. So we've built a small virtual reality simulator that is a familiarization trainer. You go ahead and you put the helmet on, and you can walk through the various tasks.

There are a large number of people who've got to be familiar with this to go and negotiate and work out problems with the mission when it happens in real time.

So we're looking for a lower fidelity trainer that can get people up to speed on the general workings of the environment that does not have a restricted point of view. What we're looking at is the use of the technology to give us an unrestricted point of view for examining the problem.

We're learning a lot from it. It's got a lot of potential – at least for the part of the work we're doing with ISD.

Another area that we're working on is intermediate level training environments. We're finding that the computer-based training is great for failure recognition and identification, but in terms of teaching coordination between people we need a different kind of environment. So that's another great area for technology environment. How do you build environments where four or five people can practice as a team, work on something significant that requires that to exercise team skills, but is smaller and not as expensive as the full-up operation?

MONTEMERLO: (Concl'd)

Real-time expert system automation: We've done a lot of work in that area in Mission Operations. We've baselined that into the next-generation control center, which we're in the process of going and building. Network management is a big challenge for us. Because almost all of our facilities have local area networks in them now. Keeping the whole smash up and running is a real challenge, because we've got computers all over the place. Finding out that they're all doing the right thing in the right way is a real challenge.

Last two big areas for technology are in software sustaining engineering and tools. As I said this morning, most of the cost of mission operations is in software sustaining and engineering. It's not in people. It's not in hardware. It's in software. It's in software in off-line tools, in real-time tools, or in simulators. So it's the major part of our budget. We're beginning to apply some automated tools to understand legacy systems in the control center, and we've got a project going in our flight design area called ROSE (reusable object oriented software environment) that we think has a lot of potential for big reductions in manpower to maintain software.

The last is electronic documentation. As Mel said, we're doing some work there. We're putting all of our Flight Data File – the onboard crew procedures – in electronic form here on the ground so that we can access them and work with them in the control center. We've already put all of our Flight Rules in electronic form in a cradle to grave process there. So that's another big technology area we're working on there.

CORNUM:

I'm Kory Cornum from the 58th Fighter Squadron. This is quite a different environment than I'm used to operating in. A fighter squadron is quite a bit different.

This is old data now – this is 3 years ago. Three years ago about right now we were getting ready to go out to Saudi Arabia, and we didn't know whether we were going to be going for a week or a month or a year. It was an unknown, and we loaded up the jets and got ready.

We deployed in late August of 1990. It was a 14-hour nonstop flight from Eglin Air Force Base in the panhandle of Florida over to Saudi Arabia. Quite an eventful flight. We usually go up over near Iceland and down through Europe, but to get us there quicker we went up to about the point there of Virginia and then straight across the water. It was terrible weather that night. We had to rendezvous with tankers all the way across the Pond, and it was fairly ugly.

CORNUM: (Cont'd)

People had a hard time, but we didn't lose any jets getting over there. We went to Tabuk Air Force Base, which is only 60 miles from Jordan. We figured our biggest threat was terrorists from Jordan, because you'll remember at that point in time Jordan and Iraq were somewhat friendly. There's not a lot of defense in that 60 miles, I'll tell you.

As soon as we got to Saudi Arabia, we started preparing for the war that just turned into Desert Shield. What we were doing over there in the first 5 months is what we call HAVA CAPs – high valuable asset protection. We were protecting AWACs. When the President and the Vice-President came over, we were flying CAPs. Those were 24 hours a day. So you may take off at 8:00 in the morning and land at noon or 1:00, and that was easy. But, if you were the guy who took off at 1:00 in the morning and were landing back about sunup, once again your circadian rhythm was somewhat disrupted. It was fairly fatiguing.

Once Desert Storm started in January, basically there were two kinds of sorties we flew. OCA (which is offensive counter air) and in offensive counter air, we're leading fighters and bombers and everybody else in to drop bombs and trying to shoot their other airplanes. My unit's strictly an air-to-air unit, so the only thing we're looking for is other airplanes. The other things we flew were defensive counter air, which continued to be the HAVA CAP and the border CAP, in case they tried to launch a big mass offensive down either into Saudi Arabia or when they were thinking about going over to Israel.

We had to maintain an alert commitment at the same time. We didn't have enough pilots to fly OCA/DCA and alert, so your crew rest, when you actually were down scheduled to sleep, you were on alert. You slept during alert, unless you got scrambled and had to go fly.

The picture on the left is some of the operational hazards. This is actually from my wife's U.S. Army unit. A camel does bad things to rotor blades, and it doesn't do anything for local Saudi relations either.

OCA or offensive counter air, those were select crews (what we call our weapons officers, guys with Ph.D.s in being a fighter pilot). A very high threat, the highly defended areas, they happened day and night. They were relatively short missions – anywhere from 2 to 3 to 4 hours. The first 2 or 3 days of the war were all planned out ahead of time. From there on it took a great deal of planning for these guys to coordinate for the bombers and the jammers and the U.S. Navy and the U.S. Marines. We don't get bombs dropped on us. You'll see the times here in

CORNUM: (Concl'd)

a minute. There's a lot of time, other than flight time, involved in these particular missions.

Defensive counter air: We just flew the CAPs again. Everybody could fly those. It was pretty low threat. It could happen day or night, and the missions were pretty long.

Here's our squadron summary. The first 15 days of the war: In January, the average pilot flew 76 hours and had 15.5 sorties. In February, the average guy flew 112 hours and had just one more sortie. So what happened is, in February, we had less sweeps and more CAPs, and so the missions got longer. We were flying anywhere from 8- to 11-hour missions up north of Baghdad. That was day after day after day, and people got quite tired. It's a little bit different than flying in the rivet joint like we learned this morning, because you're a single-seat fighter. It's a fairly demanding environment.

I think something that's very important here is motivation. The week before Desert Storm, as a flight surgeon I saw everybody there. I was seeing about 50 patients a day, with a lot of headaches and bellyaches and nervous complaints. For the first 3 days of the war, I did not have a patient. My first patient on Day 3 was somebody who hit his head on a missile and I had to sew his head up. So motivation is one of those factors that in our research and our simulations and all those kind of things you cannot put in there. I don't think there is any way to simulate it, but it is a large thing to consider when you really do go to war.

The other thing about going to war is, we play for keeps in war. You're actually getting shot at with live missiles over there; live guns. You're doing the same thing. Once again, the motivational factor in simulation, you can never think about that and actually quantify it, I don't believe.

I just got back from a week of Red Flag out at Nellis Air Force Base, Nevada. No live missiles. So we try to train as if they're alive, but in the back of your mind you know they're not. It does make things different for the operators. So in a research environment, remember that they really do shoot.

SQUIBB:

We've gone through a typical requirements implementation and operations phase here. The requirements were in part our observations of what technology challenges are. The implementation of our timeline was 50 minutes for talks; 30 minutes for discussion; 10 minutes pad; and we get out of here for drinks at 5:00. We're slightly behind that, so I'll try to get through my portion of it and leave some room for talks.

SQUIBB: (Cont'd)

Dr. Cohen, when he first opened up, within the first 2 minutes used a set of words which I think, in my mind, sets the tone for the technology challenges that are ahead of us. He used the words (referring to operations) "more efficient, more cost effective, most cost conscious." And that, indeed, is where NASA is – and I presume also the DOD partners that we have.

I want to talk about six items that I think are key to what we're going to be doing in the future. Certainly low-cost mission operations. Designing missions which are operable. The biggest leverage we have is to get involved in the design phase, as opposed to after the flight vehicles are indeed designed. I believe we'll see a merging of NASA and Defense flights. Data acquisition network capability between countries, between the United States, DOD, NASA, ESA, Russia. The uplink process is a big driver, at least in unmanned missions and I presume also in manned mission. The uplink process is currently very manpower intensive, and there are areas that we have to look at to change our concepts and the way we do work. And, technology initiatives which are certainly required in order to see the cost savings and still return the science that we have committed to do.

The cost of mission operations is not, in the eyes of NASA, acceptable currently. Perhaps a better word is that they're not affordable. The environment that we're in has changed, and we must learn how to adapt to this new environment. Missions which were designed and reviewed and approved, we had lots of comments along the way in the 1970s and 1980s are now implemented, are now in flight, are now having their operations budgets reduced significantly. Up to 30 and 40% in the years past fiscal year 1997. To make reductions of this size in missions which were already designed, which were in some ways were not operable to begin with, is extremely difficult. It doesn't mean that the mission operations were designed incorrectly. It's just that we're now in this different environment.

Whereas before we saw low risk and maximization of science as key ingredients of the way we designed our missions, they're now not as important as economy, staying within the budget, and returning an acceptable degree of science. The challenges we have are to develop new approaches which are compatible with today's financial environment. The highest leverage, as I said earlier, is to ensure that new missions are designed with the concept of minimizing the operational costs.

This we can do cheaply and easily in the design phase. It's difficult and costly once they're designed. The toughest task is to get to a mission like Galileo, an unmanned mission operation which was designed in the late-1970s, flown in the mid-1980s, and now to try to reduce the cost of that mission is extremely difficult.

SQUIBB: (Cont'd)

We must accept the concept of increased risk of losing data, but while we ensure that the risk to mission safety is not reduced. We talked a little bit earlier about the age of NASA as an organization. I believe we must empower younger engineers and scientists to perform operational tasks which are key to the success of our missions. I believe that the average age of those making operational decisions has certainly increased during the last 30 years. We must reverse this trend by bringing in younger, more inexperienced engineers and empowering them to make decisions as we did when we first joined NASA.

Manpower-intensive tasks must be replaced with automation. New processes must be devised which replace processes that have been refined. We've improved them and we've polished them, but we haven't changed them in the last 20 years. When you go about designing a mission which is operable, to me that's one in which the controllers and the users of the space vehicle think in terms of their desired observation and in terms of parameters that they use in their daily lives.

The mission design, the spacecraft design, and the operations design must be designed in a concurrent fashion. We have found that, by doing things concurrently early on in the design phase, we can indeed make tradeoffs between operations costs and spacecraft costs.

For unmanned missions, we must begin to consider the mission in terms of observations as opposed to a series of commands in a time domain. Computer programs need to be devised to transform the observations requested into the time domain if necessary. Spacecraft which are outside of the Earth orbit must be more autonomous and accept rules as opposed to accepting sequences that then they have to implement in the time domain.

We also must design spacecraft which are compatible with services which are provided by a tracking and data acquisition network: the tracking and data network, the TDRS network. The days of designing a spacecraft to optimize a set of mission parameters and then after we've designed the mission (designed the spacecraft) saying, "How much does this cost to operate it?" are gone.

I also believe that, in today's economy, we can no longer afford to launch and fly similar detectors for defense and science purposes. In the years to come, and in the near future most likely, we'll see a merging of missions in which Defense instruments will be on NASA vehicles, NASA instruments will be on Defense vehicles, and some instruments providing the same type of data will route that data to both Defense and NASA scientists. The challenge we have is to devise methods to make the combining of these heretofore incompatible and separate

SQUIBB: (Cont'd)

operations compatible and cost effective. I think the technology efforts that many of you are involved in will indeed allow us to do this.

In another arena, standards between NASA and the DOD must be agreed to – both for the downlink portion, which is in reasonably good shape, and more importantly for the uplink and the control of the spacecraft. Work is starting in the control area in both Defense and NASA groups as well as in industry, who must support these initiatives.

A space vehicle should be able to be tracked from a NASA site, a DOD site, an ESA site, and a Russian site, or from the control center in Germany that they have. We must develop and agree on a protocol which will allow for uplink and downlink standards to be used by all nations flying space vehicles. The cost of flying the space vehicle and the economy that the world is in mandates that we use these tracking resources in a way which is very cost effective as opposed to being competitive.

The uplink process must be standardized more than it has been in the past. Controlling the spacecraft vehicle is basically the same for a science flight, an engineering flight, or a military flight. The process and the protocols must be standardized so that our country's agencies which are involved can continue to gather information from space in the fiscal environment that we are now in.

Finally, I'd say that technology initiatives will be coming into the area of mission operations. The agencies which fund the technology must understand and support the fact that technology does not necessarily result in a box which one can touch, one can feel, or in a computer program which gives you a display. Much of what we need in the area of mission operation technology today is in the area of concepts, standards, and architectures which, once we agree upon these, will then allow us to effectively decide which boxes, which spacecraft programs to pursue more vigorously.

In summary, I would say that today's mission operations could be described as follows: That the operational cost which we see for the vehicles of the future must not be linear or exponential as they currently are with respect to the complexity of the space vehicle. Our space vehicles are becoming more complex and our operations are becoming more costly. This is one of the few areas where operations follows this trend. If you take a look at the aviation industry or if you take a look at the automobile industry, the airplanes and the automobiles are all becoming more complex, but the operational costs are indeed going down.

SQUIBB: (Concl'd) We must reach the era before the year 2000 when operational costs are dramatically reduced from current levels. We must use space vehicles, but we must not simulate what will happen for every single command bit that we send up to them before we send it. The technology that you are involved in will certainly help in this endeavor. But, we must ensure that the technology that we look at in the near term is focused toward lowering the number of people required to fly missions as opposed to what we've done in the 10 years before this to improve the science return and not pay attention to the number of people required in flying the missions.

KRISHEN: Really all these presentations were wonderful. The material covered here was rich and full of wealth for all of us. Is there somebody who would like to give a point of view?

SPEAKER: We've seen in the last hour and a half a very broad perspective on what needs to happen. I'd like to get the panel's perspective on what they see as the next step – the next achievable step – in accomplishing the goals that have been laid out.

KRISHEN: I'll answer it this way: The next challenge for us is to make this – for us – STIG Operations Committee really efficient. That's the challenge that I'm working to. I will let all the panelists address this question.

MONTEMERLO: That's a hard question – the next achievable step. The program that we have in artificial intelligence has been going on since 1985. About 1987, it started to have effects. I believe up to about this year, we've probably been saving as much for NASA as we've been spending. I think that will go up as time goes on. We're into a lot of different pieces of unmanned operations, crewed operations. It's hard to say what the next step is.

I think in general we need to do more specific applications. But, the next general big step is the one that I mentioned last: which is to take a look at the meta impediments to getting a lot of these things implemented, which are legal, management, and others. I guess if there was one simple answer to your question I would say it was that. Because we could do one more scheduler, or one more planner, one more knowledge-capture task, but the big one is to take the meta look at things.

I think the big thing for NASA – for sure for NASA – and I think in the space flight operations community (since I kind of worked in both DOD and NASA) is to bring the state of the practice up to state-of-the-art. Any place we go, we find that the state of practice of the way we do mission operations is at least 10 years

MONTEMERLO: (Concl'd) behind commercially available, off-the-shelf technology. We did that because we made a very conscious decision in the late 1970s and early 1980s to go ahead and put that money into Shuttle development rather than putting it into trying to bring the operations up to speed.

I think that was part of it. I think the other big part of it was because information technologies exploded. I think the big step and challenge for us as operators and for technologists is not necessarily to develop reams of brand new technologies. The problem is finding a way and an architecture where we can introduce technologies as they become developed out there in the commercial marketplace and getting them integrated rapidly.

We're never going to catch up. As soon as we'd finish making Mission Operations 30% more efficient, people are going to want to make it 45% more efficient from the 1991 baseline. So the only way to do that is to be able to ask, "What's come out there in the market? What can we go in and utilize rapidly?" So I see that as our big step. Finding a way to cut the cycle time so that we can get all of our operations up to the state of what people would consider commercial off-the-shelf.

CORNUM: I think in the operational U.S. Air Force that we'd like to think of ourselves as the tip of the spear, but probably we're the tip of the arrow. The guy that's shooting the arrows has a whole bunch of the arrows, only the tips of the arrowheads don't talk as well as they could. I think as technology improves, we need to share that knowledge so that we're not one guy talking on the phone talking to somebody else in a different part of the world, trying to organize a mission. I think there are technologies out there to make that a lot better than on a telephone.

SQUIBB: The low power, low mass technology is just now starting to emerge, and I see subsystems now that are being tested where there's 10 times less power and 100 times less weight. I think those will be put into our spacecraft, and that's going to really change the way in which we look at them.

But, the challenge is going to be to put those subsystems in that are lightweight and low power in a way that we preserve the margin such that we're not continually having to spend the time in operations up at the 90% level with all of our margins. That's where it becomes very expensive to operate the spacecraft. So as we have lighter spacecraft and lighter power, we have to maintain the operability. I think the concurrent design, as new technologies become available, of a new technology in a spacecraft (but at the same time ensuring that they're operable and that the ground folks and those who are responsible for taking care of it once it's launched are involved in it) will be key to lowering our cost.

SPEAKER: I'd like to point out that I think there's a stranglehold in all of the things you're talking about. You've got a system at the moment which is a magnificent system and that was designed 25 years ago. I think you're never going to get the cost down of doing business in space until that system's replaced. It's going to take a long time to do it.

But, I'd like just to throw out here. That what really depends on making space operations in any way competitive, efficient, or feasible is the development of a single stage to orbit vehicle.

MURATORE: I'd like to at least counter that for a second. I've read a lot about single stage to orbit vehicles. I think we've pointed out a lot of problems we have with the process and vehicles we have today. It's easy to discount what we've achieved so far and say, "This technology, which is just within our grasp, is going to provide us a lot of benefit." But, if we look at the introduction technologies, we see there's always an S-curve. That it starts out with a technology not being very productive, then it comes up the curve and it becomes very productive, then it reaches a point of diminishing returns where you can't do very much more with it.

At that point, usually you move to another technology. When you first get it, it's not as productive as the optimized technology. I think the discussions about single stage to orbit are certainly going to be very interesting to see what happens with DCX. It's certainly going to be very interesting to look at other technologies. But, if you want to go up and back today on a regular basis, there are only two options: it's a Shuttle or a Soyuz. It's well and good to say, "Gee, we can operate this by airline style operations."

I just recently read a book Dennis Jenkins wrote about Space Shuttle. All the requirements on Space Shuttle were airline-style operations. Everyone was convinced, after Shuttle, that we knew enough to build a spacecraft that could be operated like an airline. It is the exact mirror of what's being said today. The technology improvements over where we were in the late 1970s haven't been that great. I think that proponents of other activities have to be aware that the things that have dragged the Shuttle Program down in terms of cost of components and cost of operations are going to hit them, too, as soon as they hit the same scale of operations. It's going to be really interesting to see if there's been enough change in technology to effect that.

I suspect the kind of stuff we're doing with AI – with automated control centers and with electronic documentation – those are the things that are going to really bring down operations costs independent of what the vehicle is.

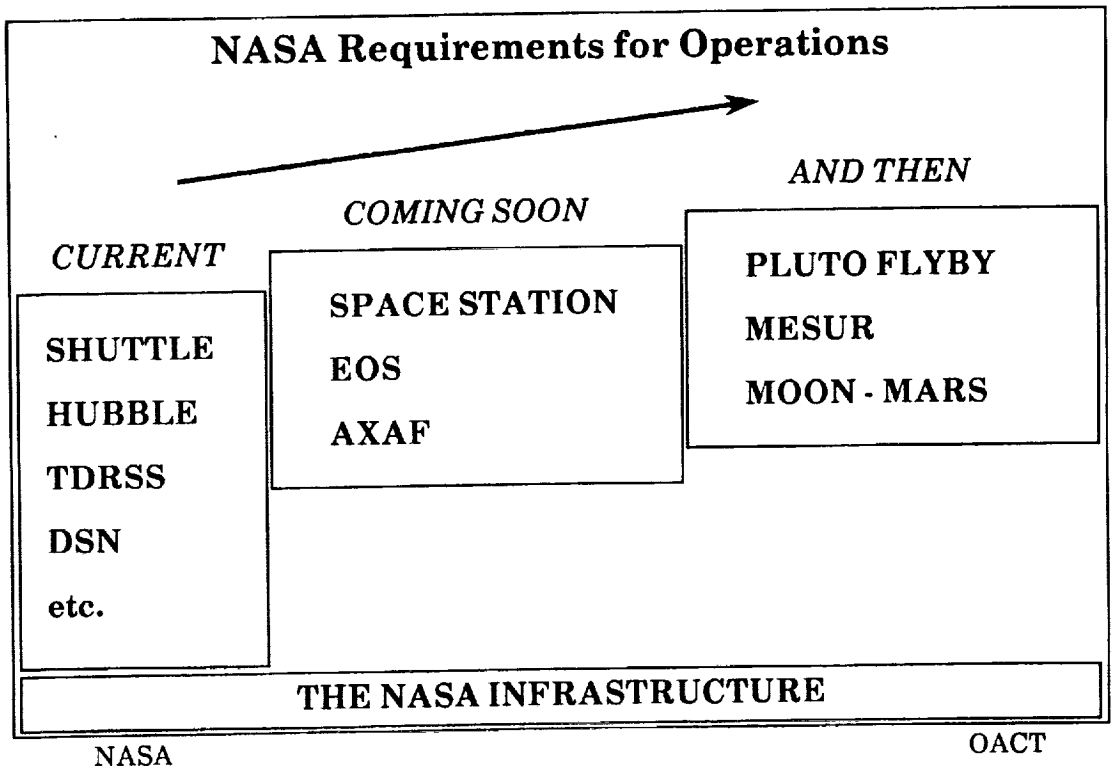
KRISHEN: Well, that's not a bad thing in a challenge situation where we have counter-points. But, I want to share one quick point of view and then I guess we'll have to close the session. My point is this: When you are optimizing something, you're saying that this is better than that or this is optimal compared to something else. You've got to have a series of things that you worked something on where you're trying to show whether it's cost, operations, or whatever: this is the minimum.

The problem in my mind is, when you compare an ongoing program with something that has not been worked on, has not been implemented, has not been manifested, it's so hard to compare what's optimum to what. So if you have a series of things that you can compare (like Apollo to Skylab to Shuttle), well there I see you should be able to compare them because they are programs that have happened. But, when you are comparing a program that will happen to something that is happening now, it's very hard to say which is going to be the optimum program. So I close the session with that comment.

Ed. Note: Vugraphs relevant to the preceding presentations appear on the following pages. If you feel your organization would like to see the videotapes from which these transcripts were taken, please let us know.

Operations Challenges at NASA

Mel Montemerlo
Manager, Operations R&D



Operations R&T Goals

- Reduce Mission *Life Cycle* Costs
  - Design, testing, and operations
- Reduce the Marching Army
  - With no decrease in capability
- More Bang for the Science Buck
  - Autonomous satellites and smart instruments
  - Fully analyze all the data
  - Let the P.I.s operate from their offices
- Productize Operations Technology
  - Transfer it to U.S. industry
- Make Future Missions Affordable

NASA OACT

Operations:
The System Life Cycle View

- Design
- Integration and Test
- Operations
- Science

NASA OACT

NASA Operations Challenges

- Manned System Operations (STS & SSF)
  - Control room automation
  - Scheduling, software engineering, training, electronic documentation
- Science Operations
  - Single person monitor/control of many spacecraft
  - Principal investigators operate from home bases
- Data Visualization and Analysis
  - Tools to easily analyze terabyte/day data rates of EOS
  - Tools to easily analyze Hubble and other images
  - Virtual reality and other tools for data visualization
  - Engineering data analysis tools

NASA OACT

NASA Operations Challenges (cont'd)

- Autonomous Spacecraft
  - Intelligent data compression for miniature spacecraft
  - Design techniques for intelligent vehicle health maintenance
  - Intelligent instruments
  - Intelligent assistance for integration and test
- Infrastructure
  - Smart forms and expert systems for procurement, payroll, personnel, logistics, insurance, travel, etc.
  - Tools to capture and utilize corporate memory

NASA OACT

USER REQUIREMENTS FOR OPERATIONS TECHNOLOGIES FOR FY1994

	TRANSPORTATION	STATION	ASTROPHYSICS	PLANETARY	SPACE PHYSICS	LIFE & M-GRAV SCI	EARTH SCIENCES	COMMUNICATIONS	INFRASTRUCTURE	
GROUND OPERATIONS										
Control room automation	X	X	X	X		X	X	X		
Launch processing operations	X		X			X				
Mission planning and scheduling	X	X	X			X	X	X		
Software Engineering	X			X						
Training	X	X				X				
DATA VISUALIZATION & ANALYSIS										
Data Analysis &Visualization Tools	X			X		X	X			
Archival and revtrieval (Hypermedia)		X				X	X	X		
Distributed P.I.s (Telescience)						X	X			
AUTONOMOUS S/C & VEHICLES										
Data compression & mgmt			X					X		
Intelligent autonomous control	X			X		X				
Intelligent instruments				X		X				
DESIGN & TESTING										
Engineering Design Tools	X	X						X		
Intelligent aids for Integration &Test	X									
INFRASTRUCTURE										
Electronic Documentation, Smart Forms									X	
Distributed Videoconferencing									X	

May 21, 1993

Meta Challenges

- Change Reward Structure to Facilitate Cost Cutting
  - Insertion of advanced technology into projects
  - Contract incentives
  - Rewards for managers to cut costs and staffs
- Commercialization of Dual-use Technology
  - Fix the contractual impediments
  - How to avoid challenges to selections
- Change Ways of Doing Business in R&D
  - Reward researchers who successfully "infuse" and commercialize
- Help Potential Users See "Multi-applicability" of Technologies
  - Do research in one application domain, but apply it to many

NASA OACT

Mission Operations Challenges

John Muratore, NASA/JSC

Challenges

- Cost Reduction
  - Maintaining safety and quality while reducing costs
- Maintaining knowledge and experience base
  - Personnel, procedures, and software
  - Accessing that knowledge rapidly when required
- Better EVA training process
  - Understanding the limitations of current water tank environment
- Aging systems—hardware and software requiring upgrades
- Improving science return

Technologies

- Global Positioning System—autonomous navigation, reduction of ground tracking requirements
  - Flight tests of GPS on STS-61 and -59
  - STS-56 flight test of handheld unit showed importance of good antenna coverage
  - JPL has had good success on TOPEX
  - Suspect GPS will be excellent for raw position determination, but computations will be required to get good velocity determinations for state vectors

Missions Operations Challenges Slide 2

Technologies (cont'd)

- Virtual Reality—potential for crew training for EVA, remote operation of spacecraft, flight controller training
  - Being used for EVA and flight controller training for STS-61 Hubble Repair
  - Resolution of Virtual Reality goggles needs improvement
- Intermediate Level Training Environments
  - Flight Controller Trainer being upgraded to handle multiple systems
  - Need intermediate level between part-task trainers and full-up mission simulations

Missions Operations Challenges Slide 3

Technologies (cont'd)

- Real-time Expert System Automation
  - Significant expertise in this area developed in mission operations
  - Expert system technology and tools are baselined into the current control center upgrade
- Network Management
  - Control center, simulation, and off-line computing environments all are based on networks
  - Network management technologies important for reducing costs, maintaining security and quality in these times

Technologies (cont'd)

- Software Sustaining Engineering Tools
  - Major part of MOD's budget is software sustaining
  - Application of automated tools to understand legacy systems being used in the Mission Control Center
  - Reusable Object Oriented Software Environment (ROSE) is critical experiment being performed in flight design areas
- Multimedia and Electronic Documentation
  - Critical tool for rapid access to design and operations knowledge rapidly
  - Electronic documentation project is working towards placing all flight data file (onboard crew procedures) in electronic form
  - Flight rules already going to all electronic cradle-to-grave process

Eagles Over Iraq: A Desert Storm Experience

Major Kory Cornum
AL/CFTO
Brooks AFB, TX 78235

August 1990 brought me the chance to deploy to Saudi Arabia in support of Operations Desert Shield and Desert Storm. As the flight surgeon for the 58 Fighter Squadron, Eglin AFB, Florida, I was right in the middle of both the planning and the actual deployment of our squadron to Tabuk, Saudi Arabia. Our squadron flies the single seat F-15C Eagle. The jet is used exclusively as an air to air asset. This paper will discuss the operational experiences we faced.

The deployment to Saudi Arabia came near the end of August 1990. The squadron deployed after several "false" starts due to higher headquarters taskings. These "false" starts took their toll both on the pilots and their families. All of us were emotionally drained after several "final goodbyes". The 14-hour flight began at 1800 hours local time. The crews flew through the worst time as far as circadian rhythm is concerned. Maintaining an alert state and not falling asleep over the Atlantic was a real problem. Most of the pilots used a stimulant to remain alert. Our route was plagued with bad weather, which made many of the multiple air to air refuelings quite difficult.

Once in Saudi Arabia the flying during Desert Shield consisted of Combat Air Patrol (CAP) missions, local training sorties, and alert. The CAP missions were to protect high value assets such as AWACS. These missions were typically 4 to 5 hours in length and occurred around the clock. The most significant missions from a fatigue standpoint were the late night/early morning missions. The missions were relatively boring sorties and a highly alert state was at times difficult to maintain.

When the war actually started the missions were either CAP, offense escort sorties, or alert sorties. The CAP sorties lengthened out to 6 to 10 hours while the escort sorties were typically 2 to 3 hours. Initially one group of pilots flew escort and another group flew CAP. Also, due to the tasking, all pilots flew both day and night as opposed to two shifts. This led to serious circadian rhythm disruption.

During the first 10 days of war one escort pilot flew 44.1 hours and 11 sorties. The individual sorties are shown below. What the flight times in the table do not show are the hours spent planning missions and the debriefing time after the missions. The sorties from the first 10 days of the war for a pilot who flew primarily CAP missions are also shown below. He flew 68.5 hours and 14 sorties. Once again most pilots used a stimulant during some of the sorties. What is significant about these numbers is that the average F-15 pilot usually flies about 15 to 20 hours a month. This represents a radical change from the standard. Also remember that in a single seat fighter you cannot stand up, take a nap, or anything that normally helps to keep one alert.

Escort Pilot

Date	Take-off Time	Hours	Mission Type
17 Jan	1400	3.8	Escort, Mig Kill!
17 Jan	2300	3.6	Escort
18 Jan	1800	3.1	Escort
19 Jan	2200	3.1	Escort
20 Jan	1200	2.1	Escort
20 Jan	1800	5.6	CAP
21 Jan	0700	6.0	Combat SAR
22 Jan	0100	2.1	Escort
24 Jan	2300	4.3	Escort
26 Jan	1700	7.8	CAP

CAP Pilot

Date	Take-off Time	Hours	Mission Type
17 Jan	0400	6.1	CAP
17 Jan	1600	8.0	CAP
18 Jan	1600	5.8	CAP
19 Jan	0700	5.0	CAP
19 Jan	1500	2.5	CAP, Mig Kill!
20 Jan	1800	5.4	CAP
21 Jan	1000	3.6	Escort
22 Jan	1700	6.2	CAP, Off Alert
23 Jan	0100	2.3	CAP, Off Alert
23 Jan	0500	7.5	CAP
23 Jan	2000	3.1	Escort
24 Jan	2400	7.2	CAP
26 Jan	1900	1.3	CAP, air abort
29 Jan	2030	4.5	CAP

The 58 Fighter Squadron flew more hours and more sorties than any squadron deployed to Desert Storm. We achieved 16 of the 32 air to air victories against Iraqui jets. Overall the deployment was great and I would do it again. The comradeship that develops in a situation like this is unique. These operational experiences are unique to wartime operations, but they illustrate the demands placed on fighter pilots.

Operations Challenges

Gael F. Squibb, NASA/JPL

Operations Challenges

- Low-cost Mission Operations
- Designing Missions Which Are Operable
- Merging of NASA and Defense Flights
- Data Acquisition Network Compatibility
- Uplink Process Standardization
- Technology Initiatives – A Hardware Box Is Not Required

Low-cost Mission Operations

The costs of mission operations are – in the eyes of NASA – excessive, or perhaps not affordable. Missions which were designed, reviewed, approved, and implemented are now having their operations budgets reduced – significantly – up to 30% in the years past FY97.

This does not mean that the mission operations were designed incorrectly. We are now in a different environment and low risk and maximization of science are not as important as economy and staying within a budget which has been reduced.

The challenge is to develop new approaches which are compatible with today's fiscal environment. The highest leverage is to ensure that new missions are designed with the concept of minimizing operational costs. The toughest task is to reduce the operational costs of missions which are in flight.

Low-cost Mission Operations (cont'd)

We must accept the concept of increased risk of losing data while ensuring that the risk to mission safety is not reduced.

We must empower younger engineers and scientists to perform operational tasks which are key to the success of missions. Thirty years ago many of us who are now in our 50s were responsible for operational control tasks. The average age of those making operational decisions has certainly increased during the last 30 years. We must reverse this trend.

Manpower-intensive tasks must be replaced with automation. New processes must be devised which replace processes that have been refined and improved and polished – but not changed in the last 20 years.

Designing Missions Which Are Operable

An operable mission is one in which controllers and users of the space vehicle think in terms of their desired observation and provide input in terms of parameters with which they are familiar.

The mission design, S/C design, and operations design must be done in a concurrent fashion. We can no longer afford to design a mission and S/C and then let the operations staff figure out how to fly the mission.

For unmanned missions we must begin to consider the mission in terms of observations as opposed to a series of commands in time domain. Computer programs need to be devised to transform the observation requests into the time domain when necessary. S/C which are outside of Earth orbit – with long flight times – must become more autonomous, accept rules as opposed to sequences, and have greater margins.

Designing Missions Which Are Operable (cont'd)

We must design S/C which are compatible with services provided by tracking and data acquisition networks. The days of designing an S/C to optimize a set of mission requirements while ignoring the operational aspects and existing standard services are gone.

Pathfinder Example

Merging of NASA and Defense Flights

In today's economy, we can no longer afford to launch and fly similar detectors for Defense and science purposes. We will see a merging of missions in which Defense instruments will be on NASA vehicles and NASA instruments will be on Defense vehicles, and the same instruments will provide data to both DOD and NASA. The challenge is to devise methods to make the combining of these heretofore separate operations compatible and cost effective.

Standards between NASA and DOD must be agreed to both for the downlink portion of data capture and transport, but more importantly for the control of S/C. Work is starting in the control area and both Defense and NASA groups must support these initiatives.

Data Acquisition Network Compatibility

A space vehicle should be able to be tracked from a NASA site, a DOD site, an ESA site, or a European national site such as GSOC. We must develop and agree on protocols which will allow for uplink and downlink standards to be used by all nations flying space vehicles.

Uplink Process Standardization

Controlling a space vehicle is basically the same for science flights, engineering flights, and military flights. The process and protocols must be standardized so that our country's agencies can continue to gather information from space in the fiscal environment we are now in. One step toward this aim is the Spacecraft Control Workshop being held in Boulder, Colorado, on August 23 through August 25.

Technology Initiatives – A Hardware Box Is Not Required

Agencies which fund technology development must understand and support the fact that technology does not have to result in a box which one can touch and feel and see. Much of what is needed in the area of mission operations technology is in concepts, standards, and architectures which will then allow one to effectively decide which boxes or S/W tools to pursue.

Summary

Today's mission operations challenges are best described in the following way.

Operations costs must not be linear or exponential with respect to the complexity of space vehicles. We must reach the era before the year 2000 where operational costs are dramatically reduced from current levels. We must use space vehicles, not simulate what will happen before we send a command. Technology will certainly help in this endeavor, but attitudes, concepts, and approaches must also change.

KEYNOTE ADDRESSES

The Current Environment for Technology Development at NASA

Gregory M. Reck*

Acting Associate Administrator, OACT

*Presented by Mr. Melvin Montemerlo

Current Environment

- Cold War End
  - National missions redefined/eliminated
  - Reduced resources
  - Great uncertainties (missions, alliances, and priorities)
- Stress on Economic Security
  - Trade imbalance
  - U.S. awakening to competitiveness issues
  - Old threats (Europe, Japan, and other Far East)
  - New threats (Russia, other FSU, and China

NASA OACT

Current Environment (cont'd)

- Clinton Administration Responses/policies
  - Conversion for economic recovery
  - Dual-use technology
  - Partnerships, cooperative programs (i.e., team processes → interdependency)
- Moreover . . .
  Expect Continual and Penetrating Oversight From:
  - Congress
  - Administration
  - Press

  For proof of cooperative, coordinated management → interdependency

NASA OACT

Quotes

"All laboratories managed by the Department of Energy, NASA and the Department of Defense that can make a productive contribution to the civilian economy will be reviewed with the aim of devoting at least 10-20 percent of their budgets to R&D partnerships with industry."

– *Technology for America's Economic Growth. A New Direction to Build Economic Strength,*
President William J. Clinton

"This new organization will be an entirely new breed – a highly flexible, customer-driven organization that will develop innovative concepts and high leverage technology that both fulfill NASA's needs and have significant commercial capabilities."

– NASA Administrator Daniel Goldin

NASA OACT

STIG Goals vis-à-vis Clinton-Gore Policy

Clinton/Gore* :	Encourage more cooperative research between federal laboratories, industry, and universities
STIG	Facilitate and encourage cooperative development programs
Clinton/Gore* :	Ensure coordinated management of technology across government agencies
STIG	Avoid duplication of effort and resources on space technology

*Taken from "Technology for America's Economic Growth," A report of the Clinton-Gore Administration

NASA OACT

Changes in Overall NASA Approach to Missions

- Increased Emphasis on Small, Quick, and Inexpensive Missions
  - Require small instrumentation payloads
- Increased Emphasis on the Infusion of New Technology as a Specific Mission Goal
- Increased Emphasis on Contribution of Technology to National Competitiveness
- Improved Coordination and Willingness of NASA Mission Offices to Use New Technology

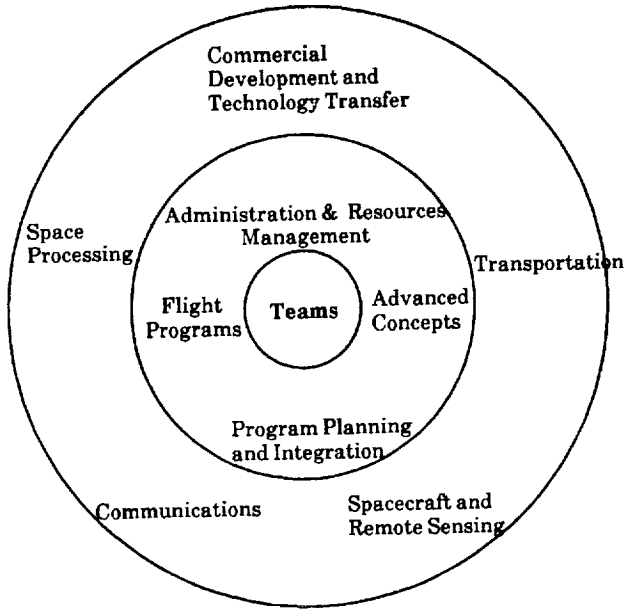
NASA OACT

OACT Mission

To pioneer innovative, customer-focused space concepts and technologies, leveraged through industrial, academic, and government alliances, to ensure U.S. commercial competitiveness and preeminence in space.

NASA OACT

OACT Organization



NASA OACT

Spacecraft and Remote Sensing

- Spacecraft Technology to Reduce Cost and Launch Weight
  - Timed and Pluto
  - Advanced concepts and designs for tenfold reduction in size and weight
- Operations Technology to Reduce Mission Cost and Enhance Productivity
  - Artificial Intelligence
  - Robotics
  - Rovers
- Advanced Instrument Technologies and Data Systems for Next-generation Observation Systems
  - Sensors and detectors
  - Optical systems
- Commercial Applications of Remote Sensing Information

NASA OACT

Strategic A&R Vision: Lower Cost and Greater Productivity

- Artificial Intelligence
  - Highly automated mission control
  - Virtual reality to reduce training costs
  - Reduce NASA infrastructure (procurement, travel, payroll, etc.)
  - Intelligent data analysis tools
  - Vehicle health monitoring and maintenance
- Rovers
  - Autonomous rovers for Mars exploration
  - Shuttle ground processing
- Robotics
  - Robotics for man-tended Space Station
  - Shuttle ground processing

With immediate commercialization and technology transfer via industry-academia-NASA teams

NASA OACT

We Have To Do Better in Space

Dr. R. Earl Good

It is wonderful to be back in my home state. My brother lives just a few miles from here. I have walked around and sat in on the conference. This is really a great conference. SOAR and STIG are very important efforts to identify and characterize interdependent space activities. You are leaders in encouraging interdependent programs and in reducing space costs. I believe that at the working level you have done an excellent job of laying out a database of who is doing what and the challenges. You are trying to maximize the resources at our disposal. Thank you for the opportunity to be with you tonight and to gain your insights in where we go next in "doing better in space."

I want to spend my 10 minutes discussing that we need to do better in space. That is obvious from hindsight. Speaking of hindsight, let me tell you what happened on my flight down.

On the flight down, my seat mate was late in arriving and in his hurry sat down on top of a newspaper. After we got into the air, he relaxed and started reading a magazine. I noticed the passenger across the aisle looking at him from time to time. Finally, she said, "Have you finished reading your newspaper?" He looked at her puzzled. But after a few moments he got up, picked up the newspaper, refolded it, and sat back down on it. He then turned to the woman and said, "My hindsight is pretty good."

Hindsight is always good. The title of my talk suggests our hindsight tells us we can do better in space. I hope we can change our methods and do better. I am going to describe where we are and to suggest some new directions.

Space is no longer the attraction to John Q. Public it once was. I remember that November time long ago when I climbed up to my dorm roof and watched Sputnik fly over. No doubt many of you have stories to tell of when you first became enthralled with space, whether it was watching John Glenn orbit or being glued to your television wondering whether the Moon lander would sink into miles of dust or land safely. Unfortunately, our children recognize space as something you do that is ordinary, sort of like going down to the corner to get a snack. Our grandchildren may take a vacation in space.

Maybe we are so jaded that it's true, as I read in a cartoon, that the "reason he's never seen a constellation is he's convinced there *really are* white lines connecting the stars." Space is something young people learn about second- or third-hand through reading books. The excitement for them doesn't exist. Their parents did that stuff.

Our political leaders' attention has now turned to the economy, as it should. With the demise of the Cold War, the need for a massive defense space capability has begun to wither away. Yes, we still need to maintain surveillance and have the capability to use the communication, missile warning, and theater awareness to ensure that the United States can prevail in any future conflict. But how do we do that and not push the U.S. deeper into debt? How can we afford the Space Station? You all are very much more up-to-date with the Congressional and Administrative maneuvers to downsize the Space Station than I. Can we really have a meaningful Space Station for these advertised low costs?

Let me now turn to another issue – *Studies*. Figure 1 shows the ever-increasing number of studies by our leaders on which is best for defense procurement. I am sure studies of space are following this trend. The newest study is the National Space Facilities Study. We are studied to death with no real changes happening. I was surprised to find a cartoon in this year's July-August edition of *American Scientist* that shows a group sitting at a conference table and the caption reads, "Washington is looking to the scientific community for an answer. Gee, I've wanted to say that my whole career." On the other hand, there really is a message here. The scientific community still may have an opportunity to get its act together and put forth an economically viable space plan.

The recently released joint National Academy of Sciences, National Academy of Engineering, and the Institute of Medicine report opens with the statement, "For 50 years, the United States poured money into basic research and subsequently reaped the rewards of that science." But the leaders of the Government's science operation are now calling for an overhaul of that system. They advocate a continued basic science, but one that takes a measure of worldwide activity and looks for the niche the U.S. does best. While the panel sees a strong Federal role in the support of basic science (quoting the July 13 Transaction of the American Geophysical Union, EOS), technology is another matter. The report states that "technological leadership in the commercial marketplace is the responsibility of the private sector" and maintains that the Government should primarily limit its role to creating "an environment in which technology can flourish" through its policies affecting such areas as investment, taxes, trade, and health and environmental regulations. However, in those commercially promising areas where research and development "may be too costly, lengthy, or too risky for an individual company," that report states that "a role for the Federal Government can make good sense." The Federal goal should be "maintaining leadership positions in those technologies that promise to have a major and continuing impact on broad areas of industrial and economic performance."

Changes can be summarized in two trend charts (figs. 2 and 3) that describe the Air Force business and, possibly to a large extent, the NASA business. Figure 2 illustrates the changes from performance-driven to cost-driven and from long development cycle to short development cycle. This is occurring because, as we see in figure 3, computers and new technology are doing much of the work – replacing people. Furthermore, the time scales have rapidly accelerated. We in Government have not adapted our "bureaucratic" processes to the new time scales. Figure 4 is a notational chart that shows why Government is always at the tail end. Our spacecraft are flying with at least 10-year-old technology. I am sure each of you has personal experiences and can tell of having a faster, more capable computer at your home in comparison to the slower and more expensive obsolete computers at work.

President Clinton in a letter to Daniel Goldin said he wants NASA's future "linked more firmly" to economic competitiveness and "long-term environmental needs" (*Aviation Week*, p. 19, July 26). Estimates that each Shuttle flight costs $1B (Roger Pielke, Jr., *Aviation Week*, p. 57, July 26) must make the general public cringe since they cannot see value returned. Last week NASA's Administrator notified the Agency's employees that the scaled-back Space Station approved by President Clinton will mean the loss of about 1300 jobs among Federal workers alone. That is about three times the number of people in the Air Force Science and Technology laboratories who accepted incentivized retirements in July 1993.

Our industries and our institutions are going to shrink during this decade. You are all aware of the base closures in DOD and the consolidation of Air Force labs. Phillips Laboratory, my lab, was formed from four smaller labs to create the Air Force "Space" lab. The national labs – Sandia, Los Alamos, and Lawrence Livermore – are worried about their consolidation.

Secretary Aspin, reviewing for industry leaders a draft of the "bottom up" study, indicated they could expect industry to shrink by the end of the decade to one manufacturer for aircraft carriers, submarines, and tanks, and two rocket manufacturers and two shipyards (*Defense News*, July 26).

Will NASA consolidate its centers? Should NASA and the Air Force together consolidate centers and laboratories? These are major questions we have been avoiding. I personally believe that we have to face the questions ourselves in a win-win situation. Otherwise we may all be on the sidelines watching others do it for us.

I am reminded of a story I heard the other day. Jim really loved his dog. He petted him before leaving for work and the dog would wait until he returned. Well, this town instituted the leash law. This meant that Jim had to tie the dog when he left for work. Jim, however, arranged to tie a long rope to a pole he planted in the middle of the backyard. He could then tie the dog to the rope and the dog could run around. It happened that before Jim left for the office in the morning, he would exercise the dog by running around the pole with the dog. After a while, just opening the back door in the morning would start the dog running around the pole in anticipation of the romp with Jim. One day Jim looked out the window and he could see the rope was ripped away from the pole; but as he went out, the dog immediately began running around the pole as if the rope were still there. The Air Force and NASA have followed a pattern and developed habits. Habits are certainly very hard to break.

I believe that we – NASA and the Air Force – have to stress our work together. Through STIG, you've made an important first step. Now we have to go beyond being aware of what each is doing. In the midst of the downsizing of our institutions, we can derive efficiencies and rebuild our programs so they will interlock like pieces of a puzzle. It is clear that NASA is responsible for man-in-space. Man-in-space is dangerous and we go through difficult and complex steps to minimize the risk. We need inexpensive, reliable space launchers for unmanned satellites (witness the Titan IV explosion). We take far too long to prepare and launch a spacecraft or shuttle. Figure 5 illustrates the months needed to prepare for a launch. We have to find ways to shorten this time if we are ever going to cut significantly the launch costs. Phillips Laboratory is working on space power, common bus architecture, environmental sensors, computers, cryocoolers, and next-generation rocket propellants. NASA is researching and developing other key components that together sustain our Nation's space capability. We need to bring it all together into one interrelated, interdependent program. Groups such as yours are one key to identifying early opportunities for cooperation that will help the Nation afford to take its rightful leading position among the spacefaring nations. Keep up the good work. Let us call on our leaders to take the next step.

Thanks again for allowing me to address your conference.

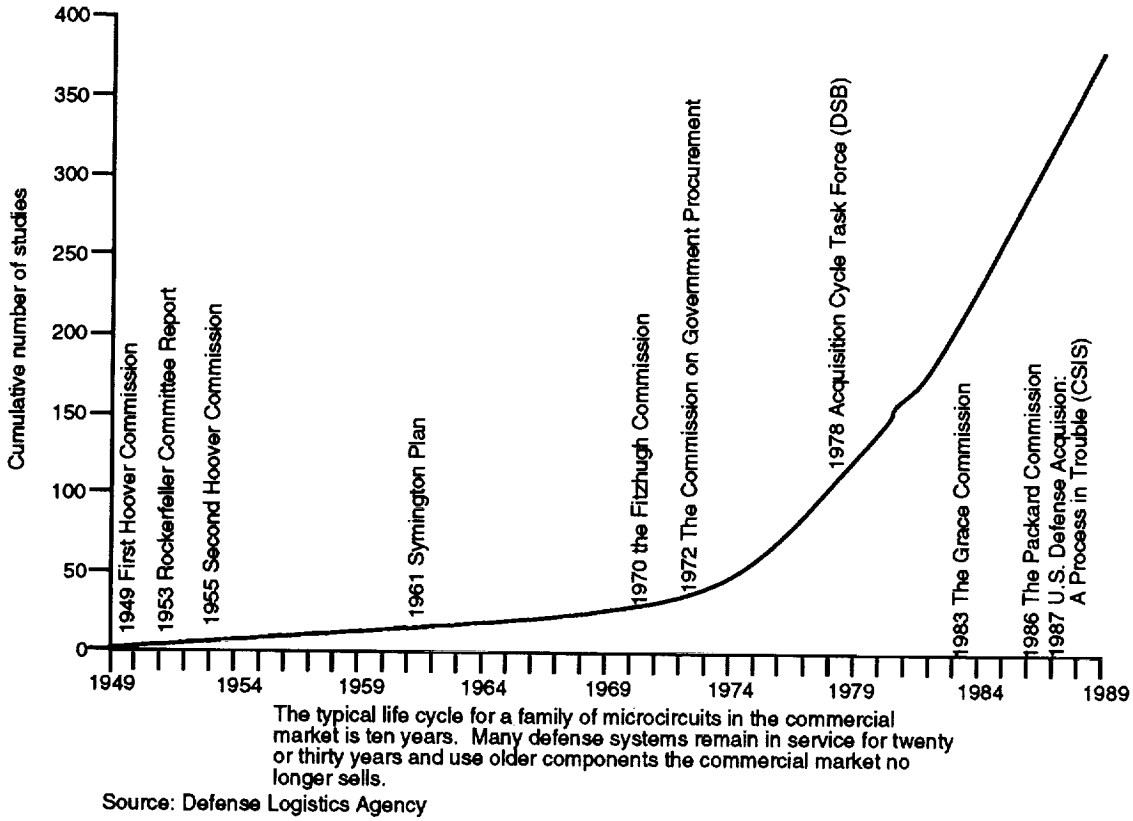

MAJOR STUDIES OF DEFENSE PROCUREMENT, 1949-1990
Cumulative number of studies
400
350
300
250
200
150
100
50
0
1949
1954
1959
1964
1969
1974
1979
1984
1989
1949 First Hoover Commission
1953 Rockerfeller Committee Report
1955 Second Hoover Commission
1961 Symington Plan
1970 the Fitzhugh Commission
1972 The Commission on Government Procurement
1978 Acquisition Cycle Task Force (DSB)
1983 The Grace Commission
1986 The Packard Commission
1987 U.S. Defense Acquisition: A Process in Trouble (CSIS)
The typical life cycle for a family of microcircuits in the commercial market is ten years. Many defense systems remain in service for twenty or thirty years and use older components the commercial market no longer sells.
Source: Defense Logistics Agency

Figure 1.

TRENDS

OUT	IN
Superpowers	Global
Strategic	Tactical
Military	Economic, Environmental
Performance-Driven	Cost-Driven
Static Environment	Time-Driven
Firepower	Information Power
Mechanical	Digital
Serial Development	Concurrent Engineering
Long Development Cycle	Short Development Cycle
Build and Test	Simulate
Train	Simulate
R&D	R&D

Figure 2.

UNDERLYING TRENDS

1) Machines are now cheaper than people / Information is now cheaper than machines

- Response: Information is replacing people and machines
  - Cars, planes, tanks, optical systems, structures, commerce, warfare, . . .
- Response: Changing paradigms
  - Information systems offsetting cost of mechanical parts
  - Much more relative investment in modeling, communicating, storing, reusing information
  - Automated manufacture

2) Time scales are accelerating

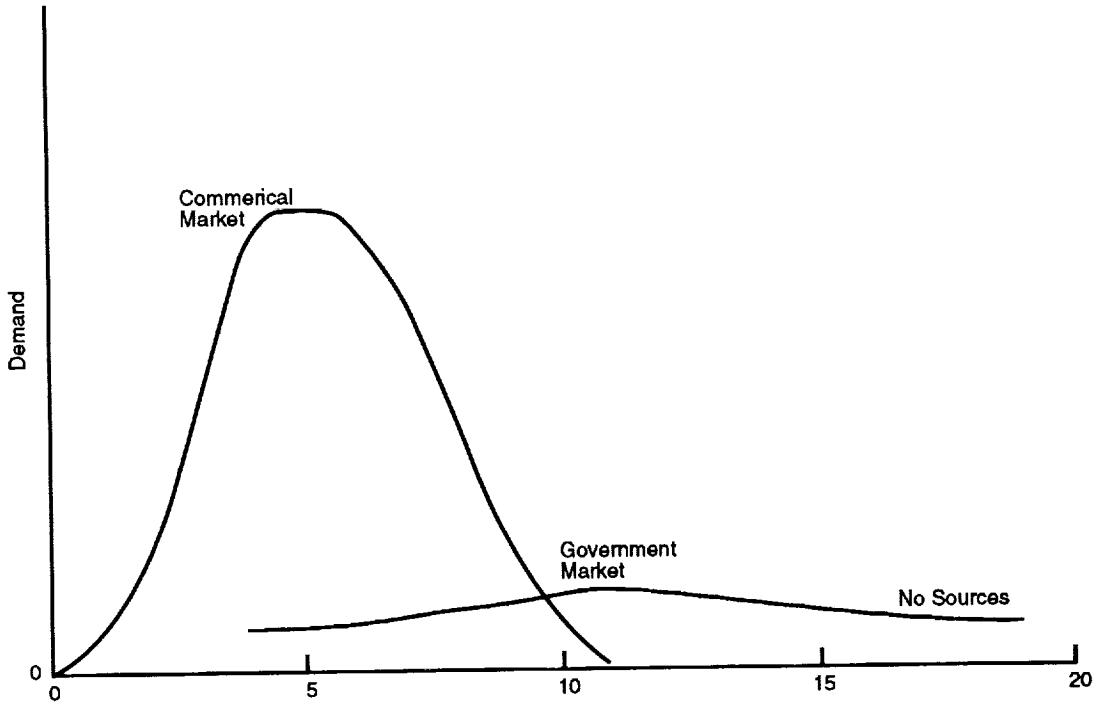
- Things lose relative value quickly (Tried to sell your old computer lately?)
- Response: Manufacturers are accelerating development cycle
  - Time-to-market is becoming the criterion of success
  - The longer you wait, the more it costs and the less it's worth
  - Concurrent engineering, modular design allow parallel development, reuse

3) Complexity is increasing

- Cars / appliances / software / telephones
- Leads to "bit systems" or "N-squared" problem
- Response: Software, hardware, manufacturing methods, and organizations are moving from hierarchical to distributed, autonomous, flat systems

Figure 3.

AFTER THE MARKET ON MICROCIRCUITS



The typical life cycle for a family of microcircuits in the commercial market is ten years. Many defense systems remain in service for twenty or thirty years and use older components the commercial market no longer sells.

Source: Defense Logistics Agency

Figure 4.

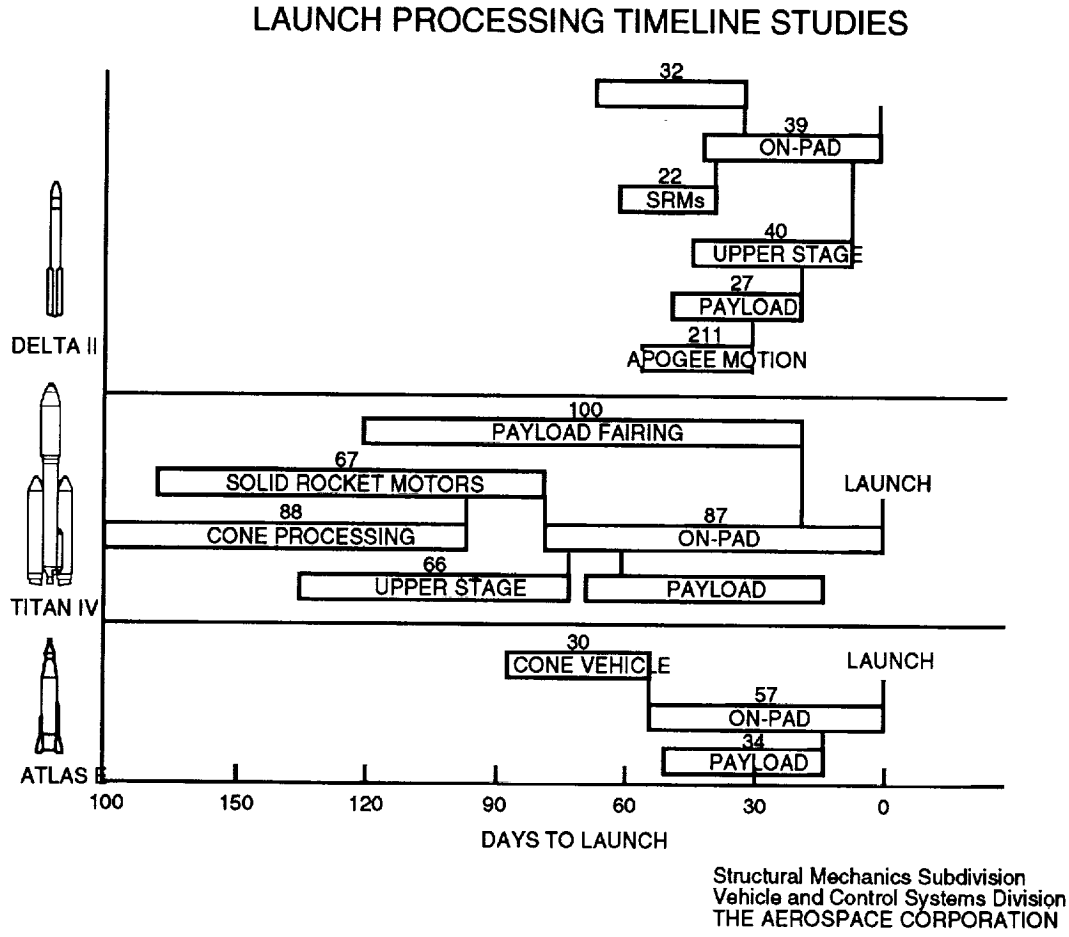

LAUNCH PROCESSING TIMELINE STUDIES
DELTA II
32
39
ON-PAD
22
SRMs
40
UPPER STAGE
27
PAYLOAD
211
APOGEE MOTION
TITAN IV
100
PAYLOAD FAIRING
67
SOLID ROCKET MOTORS
LAUNCH
88
CONE PROCESSING
87
ON-PAD
66
UPPER STAGE
PAYLOAD
ATLAS E
30
CONE VEHICLE
LAUNCH
57
ON-PAD
34
PAYLOAD
100
150
120
90
60
30
0
DAYS TO LAUNCH
Structural Mechanics Subdivision
Vehicle and Control Systems Division
THE AEROSPACE CORPORATION

Figure 5.

SECTION I

ROBOTICS AND TELEPRESENCE

Session R1: NAVIGATION, MACHINE PERCEPTION, AND EXPLORATION

Session Chair: Dr. Brian Wilcox

N94- 34020

Microrover Research at JPL

Brian Wilcox
NASA Jet Propulsion Laboratory
Pasadena, CA

Planetary exploration with microrovers can be extended beyond the baseline short-range capability proposed for the Mars Environmental Survey (MESUR) microrover mission. The useful range of the microrover can be increased by using local landmarks to accurately reach desired science sites with a minimal number of Earth commands. Furthermore, onboard processing can give indications of excessive sinkage or slippage, which comprise hazards which may not be detectable from imaging, as well as to improve dead reckoning performance. Lastly, it is important to estimate the mean and variance of mission parameters such as time-to-reach-goal, energy-to-reach-goal, and likelihood-of-finding-landmark. The author of the paper discusses research focused on accomplishing these objectives.

N94- 34021

Design of the MESUR/Pathfinder Microrover

Henry W. Stone
NASA Jet Propulsion Laboratory
Pasadena, CA

The use of unmanned robotic vehicles to assist in the exploration of Mars and other planets has been of interest to the National Aeronautics and Space Administration (NASA) for several decades and has been the focus of an ongoing research program at the Jet Propulsion Laboratory (JPL) for a similar period of time. As a result of these research activities, JPL is in the process of designing and building a small (7-9 kg) microrover to be flown aboard the Mars Environmental Survey Mission (MESUR)/Pathfinder spacecraft, which is tentatively to be launched to Mars in late 1997. The microrover will perform a variety of technology experiments designed to provide information critical to the design of future planetary rovers. In addition, the microrover will perform several science and lander related experiments using specialized onboard instruments. To enable the microrover to perform these experiments at selected target areas and at the same time deal with the long time delays (and limited communications bandwidth), a control/navigation approach combining the use of operator-designated waypoints and onboard behavior control has been adopted. The design of the MESUR/Pathfinder microrover and the overall manner in which it is controlled are described herein.

N94- 34022

Air Force Construction Automation/Robotics

Al Nease
1Lt. Christopher Dusseault
WL/FIVCO-OL
Tyndall AFB, FL

The Air Force has several unique requirements that are being met through the development of construction robotic technology. The missions associated with these requirements place construction/repair equipment operators in potentially harmful situations. Additionally, force reductions require that human resources be leveraged to the maximum extent possible and that more stringent construction repair requirements push for increased automation. To solve these problems, the U.S. Air Force is undertaking a research and development effort at Tyndall AFB, FL, to develop robotic construction/repair equipment. This development effort involves the following technologies: teleoperation, telerobotics, robotic vehicle communications, automated damage assessment, vehicle navigation, mission/vehicle task control architecture, and associated computing environment. The ultimate goal is the fielding of robotic repair capability operating at the level of supervised autonomy. The authors of this paper will discuss current and planned efforts in construction/repair, explosive ordnance disposal, hazardous waste cleanup, fire fighting, and space construction.

N94-34023

Lunar Exploration Rover Program Developments

P. R. Klarer

Advanced Vehicle Development Department
Robotic Vehicle Range
Sandia National Laboratories
Albuquerque, New Mexico

Abstract

The Robotic All Terrain Lunar Exploration Rover (RATLER) design concept began at Sandia National Laboratories in late 1991 with a series of small, proof-of-principle, working scale models. The models proved the viability of the concept for high mobility through mechanical simplicity, and eventually received internal funding at Sandia National Laboratories for full scale, proof-of-concept prototype development. Whereas the proof-of-principle models demonstrated the mechanical design's capabilities for mobility, the full scale proof-of-concept design currently under development is intended to support field operations for experiments in telerobotics, autonomous robotic operations, telerobotic field geology, and advanced man-machine interface concepts. The development program's current status is described, including an outline of the program's work over the past year, recent accomplishments, and plans for follow-on development work.

Introduction

Sandia National Laboratories' Robotic Vehicle Range (SNL/RVR) has been developing mobile robotic systems for a variety of DOE and DoD applications since 1984. Beginning in 1989, the SNL/RVR began exploring civil space applications which could make use of the existing technology base, particularly in lunar exploration missions. A philosophy that stresses simplicity in the design and implementation of a rover system wherever possible has been the basic tenet of the SNL/RVR's approach to the problem of lunar exploration. In line with this philosophy and without official funding, an innovative concept for a simple, agile lunar rover vehicle was developed and evaluated in the form of several scale models [1,2]. The Soviet Union's space program successfully operated two lunar rovers in the early 1970's [3,4] using very simple technology, thereby demonstrating that teleoperation is a viable technique despite the inherent Earth-Moon communication time delay, and that relatively simple mechanisms can provide a useful level of capability to perform meaningful science through telerobotics. Figure 1 shows one of the early models of Sandia National Laboratories' Robotic All Terrain Lunar Exploration Rover (RATLER), the focus of Sandia's lunar exploration efforts, during field testing at Death Valley National Monument in late spring of 1992.

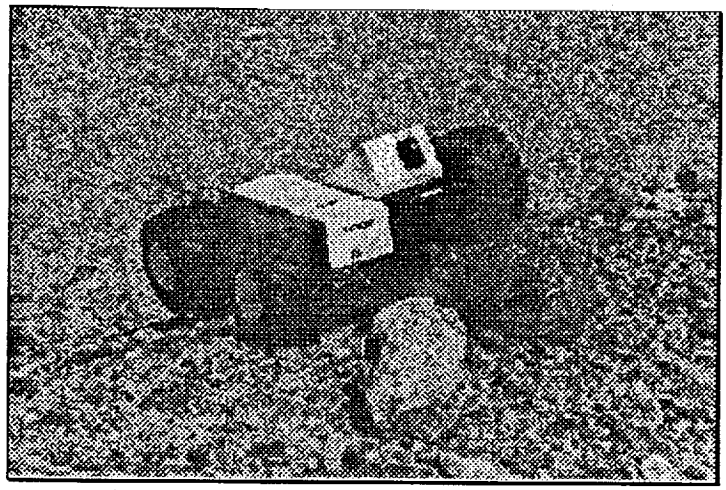


Figure 1. RATLER Testing at Death Valley

Over the summer of 1992, two summer students employed at the SNL/RVR designed, constructed, and tested a more robust version of the scale model RATLER, called RATLER-A. RATLER-A and the original models provided additional testing opportunities at the White Sands National Monument, where the RATLER design concept showed promise for very good mobility and agility characteristics in very dry, loose gypsum sand. Two additional models were built to support demonstration of the concept to NASA, DOE, and the public at the National Air and Space Museum's Planetary Rover EXPO in September 1992. Figure 2 shows the RATLER-A being operated over a simulated Mars terrain at the Planetary Rover EXPO.

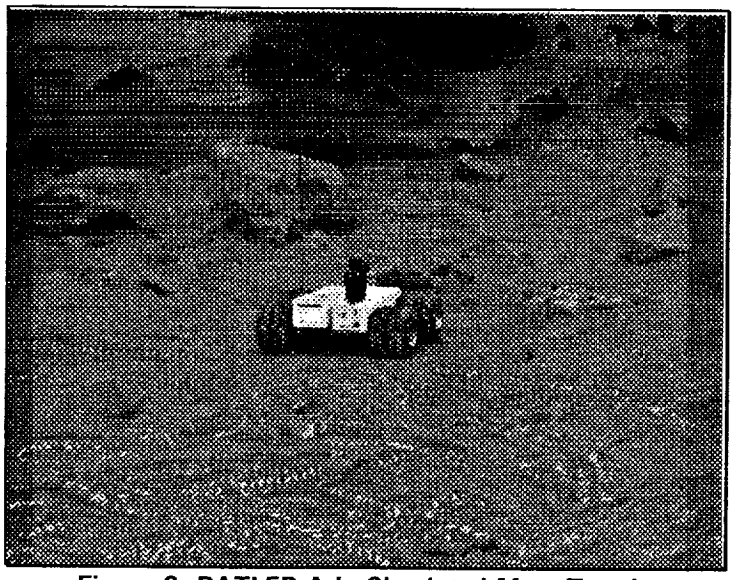


Figure 2. RATLER-A in Simulated Mars Terrain

As a result of the work with the scale models, a Laboratory Directed Research and Development (LDRD) program was initiated to develop a full scale RATLER vehicle. The LDRD project was originally proposed for a period of two years, beginning in October 1992, and was recently approved for further development in FY 1994. The remainder of this paper focuses on the LDRD program for development and testing of the full scale RATLER, called RATLER II.

RATLER II Development Program

The goals for the RATLER II development program are to develop a 1-meter scale RATLER vehicle using off the shelf technology, and to demonstrate a capability commensurate with stated or inferred requirements for a lunar exploration rover vehicle. In conjunction with the actual vehicle platform, a compact, portable Control Driving Station (CDS) is also under development to support field operations. Both the CDS and the RATLER II incorporate multiple processors on a 32 bit communication bus, and implement a real-time, event-driven multitasking software architecture.

When the RATLER II program initiated in October 1992, the first task was to determine what performance requirements or specifications existed in the literature for a lunar exploration rover. Although examples of lunar roving vehicles were found [3,4,5], a contemporary set of requirements for future missions by rovers to the Moon were not found. A trade-off study [6] was performed to attempt to derive requirements that could then be used by the project team to design and build the RATLER II. Results of that study led to a RATLER II design that could be constructed using off the shelf technology, and which was expected to meet a reasonable set of performance criteria in terms of mobility and payload capacity. The current RATLER II configuration was sized to meet the mass and volume constraints imposed by the ARTEMIS Common Lunar Lander [7], and to provide a significant science payload capacity. Figure 3 shows the current RATLER II configuration.

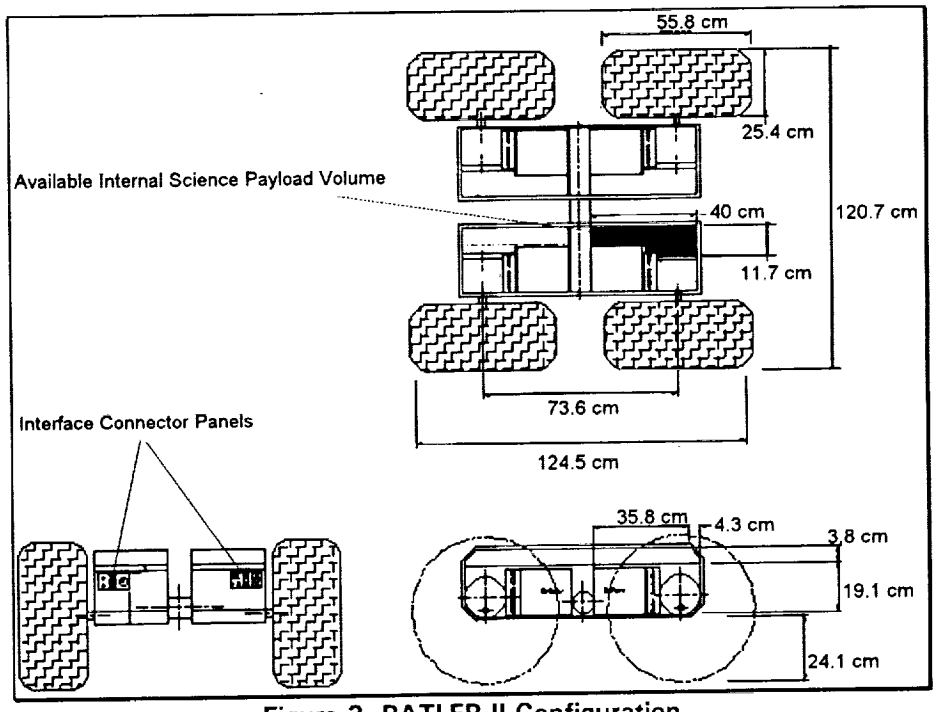


Figure 3. RATLER II Configuration

Based on the trade-off study results, a RATLER II pathfinder test article was constructed and tested at both the SNL/RVR, and at the White Sands Missile Range (WSMR) during November and December of 1992. Those field trials and additional analysis led to a few minor changes in the vehicle's configuration, which should result in improved mobility and an increase in mechanical strength of the structure. The changes included the addition of aluminum skid plates to protect the under-sides of the carbon composite chassis, larger wheels, increased drive motor torque, and a slight increase in the vehicle's lateral stance. The RATLER II prototype currently under construction is shown in Figure 4.

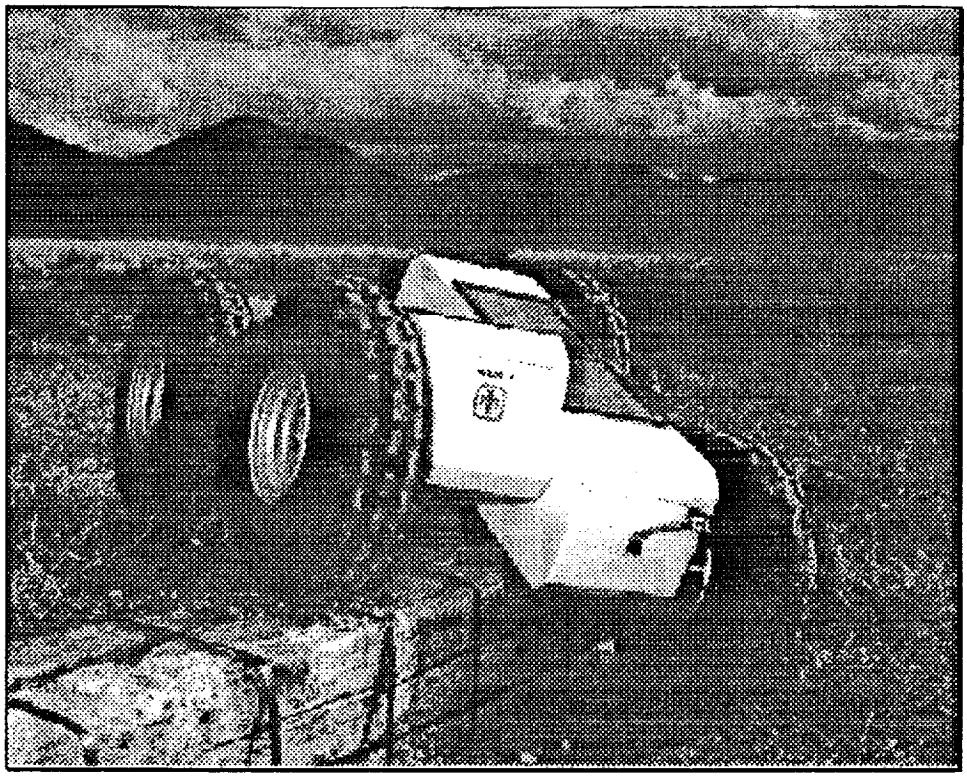


Figure 4. RATLER II Prototype

The RATLER II chassis consists of two bodies, connected by a passive central pivot aligned along the lateral axis of the vehicle. The bodies are constructed of an inner and outer skin of carbon fibers embedded in an epoxy matrix, laid over a cellulose honeycomb inner core. Each body is approximately 25 centimeters wide by 25 centimeters deep by 92 centimeters long, and masses approximately 3.2 kilograms empty. The complete system (not including science instruments) is projected to mass ~70 kilograms, including four lead-acid batteries and four rubber tires on steel rims. Table 1 lists the RATLER II's specifications and expected performance parameters.

Table 1. RATLER II Specifications

Parameter	Value	Units
Wheel Radius	28	cm
Wheel Width	25	cm
Wheelbase	72.4	cm
Stance (to center of contact patch)	81	cm
Total Vehicle Mass (TVM, no payload)	70	kg
Total Stored Volume (TSV)	0.6	$meters^3$
Maximum Single Dimension of TSV	122	cm
Maximum Speed	0.6	meters/second
Slope Stability	>45	degrees
Slope Climbing	~30	degrees
Obstacle Climbing	~75	cm
Maximum Payload Mass (additonal to TVM)	18	kg
Maximum Payload Power (planned)	100	watts (electric)
Maximum Internal Payload Volume	9600	cm^3

The drive system uses four wheel independent electric drive from four 24 volt DC permanent magnet gearhead motors, each of which provides ~22 Newton-meters of torque, and should provide a maximum speed of ~60 centimeters per second. The battery system is augmented with commercial photovoltaic arrays to provide a trickle charge capability, and is expected to provide ~6 hours of operation assuming a 50% duty cycle on the drive system. An internal payload space of ~9600 cubic centimeters and a maximum of 18 kilograms additional mass budget is provided for scientific instruments, which are allowed a total of up to 100 watts of on-board power.

The computing system being implemented on RATLER II is a commercial STD-32 system, which is based on the popular STD 80 backplane design but has been expanded to allow 32 bit data transfers. The STD-32 system supports multiple processors using a master/slave arrangement with bus arbitration and peripheral sharing support. The master processor is an Intel 80486 based machine equipped with 8 Mbytes of RAM and 1 Mbyte of EEPROM, and the single slave processor is an NEC V53 (80286/80386 clone) equipped with 1 Mbyte of RAM. Extra card slots have been budgeted to allow additional slave processors for future expansion. Shared peripheral devices on-board include a high speed, 12 bit, 32 channel Analog to Digital (A/D) converter, a 12 bit, 8 channel Digital to Analog (D/A) converter, Ethernet adapters, and a custom designed, 12 channel digital quadrature encoder board. Each of the two CPU's have on-board I/O ports which give the system a total of 5 serial (RS-232) ports and 72 Parallel Interface Adapter (PIA) ports, of which 24 are optically isolated. On-board sensors and instrumentation include a magnetic fluxgate compass, a Global Positioning System (GPS) receiver, pitch and roll axis inclinometers, an angular rate sensor for the yaw axis, a body-pivot angle encoder, individual wheel odometers, drive motor tachometers, drive motor temperature sensors, drive motor current monitors, battery voltage sensor, and a computer module temperature sensor. All of the internal components are mounted on removable payload module base plates, to allow easy access for maintenance or repair. Communications with the CDS during field operations are handled through a 4800 BAUD, full duplex digital RF modem, and an RF video/audio transmitter.

The Ethernet ports are used for development, and access a LAN at the SNL/RVR for software development tools and source code, so that code development can be accomplished directly on the target CPUs on-board the vehicle. The software architecture for each CPU incorporates a real-time, event driven, multitasking system, is written in C and C++, and accomplishes inter-CPU communications through dual ported RAM. The software system has been designed to allow future expansion of autonomous capabilities, and rapid prototyping of new experimental configurations for robotic control. Current program plans call for an initial operational capability demonstration of teleoperation in September 1993, with future work in FY94 to include the addition of autonomous navigation features.

Future Work

A major focus of the project team's efforts in FY94 will be the conduct of field trials with the RATLER II and its CDS. As noted above, a payload bay area has been allotted to carry scientific instruments weighing up to 18 kilograms and requiring up to 100 watts of power. The RATLER II program is intended to be a testbed for robotic lunar exploration, and as such provides mobility for the true focus of such a mission, i.e. the science package. Although the SNL/RVR is not developing any science packages for lunar exploration, we are offering essentially a 'free ride' during our ongoing field trials to developers of such instruments. We will provide the appropriate interface information to qualified instrument developers, to allow them access to RATLER II's support systems. With proper planning and coordination

between the developer and the RATLER II project team, integrating the science package should be a relatively straightforward 'strap-down' process, and should allow several different science packages to be operated on-board the RATLER II during field operations over the course of FY94 (through September 1994). Each proposed payload will be evaluated on an individual basis, and support funding (if any) will be negotiated as required between the SNL/RVR and the instrument developer. As long as no significant modifications to the RATLER II hardware or software is required to support the instrument, no support funding to the SNL/RVR will be required from the instrument developer.

As noted above, one of the major efforts beginning in October of 1993 will be the extension of the RATLER II's navigation capabilities to include some autonomous features. Current plans call for a subsumption-like architecture [8,9], which will also necessitate the addition of obstacle detection sensors. Various configuration options are under consideration, and it is hoped that at least two different implementations will be developed and evaluated over the course of the RATLER II program.

A six degree-of-freedom manipulator is planned for FY94, and will be among the first tasks undertaken beginning in October 1993. A dedicated slave CPU will allow coordinated motion of the manipulator while the vehicle is in motion, with virtually no impact on other on-board processing tasks taking place. This capability will allow the entire system to act as a multi-degree-of-freedom (redundant) mobile manipulator, and should provide a useful platform for field trials and testing of planetary exploration mission scenarios. An initial payload lift capacity of ~2 kilograms at full arm extension is planned, as is a small suite of interchangeable end effectors.

The current video RF transmitter incorporates two sideband audio channels, which may be used to bring back stereo audio from the RATLER II to the CDS. Although the Moon has no atmosphere and therefore sound does not travel beyond the surface (however it does travel through the Lunar interior), potential terrestrial applications for the RATLER II could make use of such a feature and we plan to incorporate it. In addition, a set of stereo video cameras will be installed along with a duplexing system to allow stereo vision over a single RF transmitter. The use of a duplexer has been implemented previously at the SNL/RVR for this purpose, and has proven to be quite effective in improving perception without the penalty of doubling the bandwidth required for transmission of the real-time images.

Another item of interest for future work in the RATLER II program will be multi-vehicle control. A second RATLER II prototype will be constructed (essentially a twin of the first unit), and will be used to explore the advantages and disadvantages of simultaneously controlling more than one rover from a common control station, by a single operator. This issue is relevant to the argument that the use of robotic rover vehicles for lunar exploration makes sense, both economically and technically.

Obviously, the wheels, solar panels, computers, and batteries being used on the RATLER II are not types which would be suitable for a space qualified system. Conceptual designs for lunar-type wheels will be explored to the extent that at least one set of wheels will be constructed and evaluated, but a comprehensive program of wheel design is not currently planned. The subject of wheel design for lunar roving machines has been explored in some detail [10], and if incorporated in this development program might easily consume the entire budget. Trade studies may be done with regard to batteries, solar cells, and computing technologies, to identify space qualified (or qualifiable) systems, but the RATLER II prototype currently under development will remain Earthbound. It is intended that a space qualified, flight-ready system could be developed based on the RATLER II, if such a program was determined to be in the national interest, but that is beyond the scope of the RATLER II program as it is currently defined.

Summary

Sandia National Laboratories' Robotic Vehicle Range has brought the Robotic All Terrain Lunar Exploration Rover (RATLER) program from an initial concept to a full scale working prototype in ~19 months. The RATLER II is designed to provide mobility characteristics and payload capacity that are sufficient to realistically demonstrate lunar exploration activities by a mobile robotic vehicle, and is sized to be compatible with payload constraints imposed by the ARTEMIS Common Lunar Lander. The RATLER II prototype itself is not intended to be a space qualified system, but should provide design and engineering data which could be used in the future for a flight qualified lunar exploration rover. The RATLER II will be operational by the end of September 1993 in a teleoperation mode, and will begin field trials in October 1993. Activities planned for the remainder of 1993 and through September 1994 include the addition of a manipulator arm, additional sensing capabilities, autonomous behavioral control software, and field demonstrations of the system in a realistic environment. Developers of science instruments that could make constructive use of the RATLER II's mobility and manipulation characteristics are invited to contact the author to discuss cooperative field trials and demonstrations of their systems, carried as a payload on the RATLER II.

Acknowledgments:

The author would like to acknowledge the many individuals who have directly or indirectly contributed to the RATLER project: Jim Purvis and Kent Biringer, coinventors of the original concept; Adan Delgado, Leon Martine, and Patrick Wing, Sandia summer students who constructed and tested several of the prototypes; Wendy Amai, Roger Case, and Bryan Pletta who constitute the current project development team at Sandia National Laboratories; and finally our many colleagues at NASA, whose comments, constructive criticisms, enthusiastic encouragement and support have greatly influenced the RATLER development.

References:

[1] PURVIS, J., and KLARER, P, "R.A.T.L.E.R.: Robotic All Terrain Lunar Exploration Rover," Sixth Annual Space Operations, Applications, and Research Symposium, Johnson Space Center, Houston, TX, (1992).

[2] KLARER, P. and PURVIS, J., "A Highly Agile Mobility Chassis Design for a Robotic All Terrain Lunar Exploration Rover," American Nuclear Society's Fifth Topical Meeting on Robotics and Remote Systems, Knoxville, TN, (1993).

[3] ALEXANDROV, A.K., ET. AL., "Investigations of Mobility of Lunokhod I", Space Research XII -- Akadamie-Verlag, Berlin (1972).

[4] FLORENSKY, C.P., ET. AL., "The floor of crater Le Monier: A Study of Lunokhod 2 data", Proceedings of the 9th Lunar and Planetary Scientific Conference (1978), pp 1449-1458.

[5] COSTES, N.C., ET. AL., "Mobility Performance of the Lunar Roving Vehicle: Terrestrial Studies - Apollo 15 Results", Marshall Space Flight Center, NASA Technical Report #NASA TR R-401, December 1972.

[6] KLARER, P., "Design and Configuration Constraints for a Robotic All Terrain Lunar Exploration Rover", Sandia National Laboratories Internal Technical Report (1992)

[7] HOFFMAN, S.J. and WEAVER, D.B., "Results and Proceedings of the Lunar Rover/Mobility Systems Workshop", Exploration Programs Office document #EXPO-T2-920003-EXPO, NASA Johnson Space Center, Houston Tx, (1992)

[8] BROOKS, R.A., "A Robust Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, vol RA-2#1, (1986).

[9] MILLER, D.P., "Rover Navigation Through Behavior Modification," Fourth Annual Space Operations, Applications, and Research Symposium, Johnson Space Center, Houston, TX, (1990).

[10] CARRIER, D., "Lessons Learned from the Lunokhod & Lunar Roving Vehicle," contained in: "Presentations from the Lunar Rover/Mobility Systems Workshop". S.J. Hoffman, D.B. Weaver, Exploration Programs Office document #EXPO-T2-920004-EXPO, NASA Johnson Space Center, Houston Tx, (1992)

Session R2: ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES: PANEL PRESENTATIONS

Session Chair: Capt. Paul Whalen

Robotics & Telepresence Research Challenges: Panel Presentation

Dr. Chuck Weisbin, NASA/JPL

Planetary Rover Challenges

Programming Thrusts

- Code S Concurrence on Needs
- Alliances with Industry and the Universities
- International Collaboration (e.g., Russia, France)
- Lunar and Venus Exploration Options

Planetary Rover Challenges (cont'd)

Technical Thrusts

1. Real-time perception and goal identification
2. Onboard placement of science payloads and rock coring
3. Sparse terrain mapping
4. Systematic benchmark experiments (e.g., legs versus wheels)
5. Fault tolerance and error recovery
6. Autonomous navigation over the horizon

In-space Robotics Challenges

Programmatic Thrusts

- Flight Experiments
- Terrestrial Demos > Commercialization
- Alliances with Industry and Universities
- International Collaboration (e.g., JPL/MITI)
- Microtechnology (In-situ Spacelab Experiments)

In-space Robotics Challenges (cont'd)

Technical Thrusts

1. Automated operation of remote dexterous robots from ground
2. Compilation and concatenation of robot skills
3. Instrumented end effectors with improved dexterity
4. Object verification and pose refinement
5. Sensory skins for obstacle avoidance
6. Safe and robust control of manipulator/environment interaction (e.g., compound manipulators, fault tolerance)

Major Technology Challenges for DoD UGV program

1993-2000

Charles M. Shoemaker
Chief, Focus Program Office
for Advanced Automation and Robotics
Army Research Laboratory

Basic Premise:

Reductions in manpower without reductions in responsibility will result in increased DoD emphasis on supervisory control modality for UGVs.

Challenges:

- **Supervisory Control of UGV's: Mission and Mobility.**
- **Optional Robotic Functionality for Manned Systems.**
- **Innovative Mobility Platform Technology.**

Superisory Control of UGVs

Motivators:

- **Minimum 60 megabit data rate for single video downlink in teleoperation mode. Requires data link in spectral region for which beyond line of sight propagation is problematic.**
- **Fiber Optic Data Link causes severe operational constraints.**
- **Multiple vehicle operation in high data rate mode causes frequency allocation problems.**
- **1-on-1 teleoperation requires increased manpower**

Superisory Control of UGVs

Technical Challenges:

- **On-board autonomy: mission function/mobility.**
- **Data compression-reconstitution.**
- **Reconfigurable Man Machine Interface.**

Supervisory Control of UGVs

Challenges (cont.)

Data Compression-Reconstitution

- **Fractal Compression.**
- **Pyramidal Compression.**
- **DCT.**
- **Foveation.**

Supervisory Control of UGVs

Challenges (cont.)

Limited Autonomous Mobility (near term)

- **Retrotraverse.**
- **CARD.**
- **Leader Follower.**
- **Road Following.**

Supervisory Control of UGVs

Challenges (cont.)

Mission Function Automation

- **Target Cueing.**
- **Target Detection Static and Mobile.**
- **Leveraging Strategy.**

Supervisory Control of UGVs

Challenges (cont.)

- **Reconfigurable Man Machine Interface.**
- **Requirement for OCU to operate both as a stand-alone and in various vehicle mounted configurations.**
- **Major emphasis on low power, flat panel displays; interface to helmet mounted displays; and synthetic binaural audio cueing to the operator.**

Optional Robotic Functionality for Manned Systems

Motivators:

- Large DoD investment in manned systems, parts, and training.

- Now, specialized robotic platforms are difficult to field at this time, must compete with manned systems for scarce airlift, and have received only luke warm military acceptance at best.

- Optional robotic functionality offers low introduction cost and the opportunity to save lives in hazardous missions. It is a useful way to introduce robotics to the military community and explore possible new mission role (e.g. decoy).

Optional Robotic Functionality for Manned Systems

Technical Challenges:

Optional robotic function design requirements

- Non-intrusive actuation and control packages.
- Minimum volume.
- Low power consumption.
- Rugged, reliable and maintainable.
- Quick disconnect/back-drivable.
- Built-in diagnostic functions.

Innovative Mobility Platform Technology

Motivators:

- **Loss of driver's "seat of the pants" sense of feel regarding wheel slip, vehicle position and estimate of obstacle size results in a near-term loss of mobility compared to manned systems.**

- **Unconventional platforms may offer a means to compensate for this mobility loss.**

Innovative Mobility Platform Technology

Technical Challenges:

- **Stability.**
- **Recovery from roll-over.**
- **Power consumption.**

Depot Telerobotics:

The Challenges

M. B. Leahy, Jr., Ma., USAF, Ph.D.

Robotics and Automation Center of Excellence
San Antonio Air Logistics Center
Technology & Industrial Support Directorate
Advanced Process Technology Section

Background

- Depot Environment
- Race Mission
  - Command Focal Point
  - Technology Pull
  - Champion

Background (cont'd)

- Motivation: Judicious Tech Insertion
- Paradigm:Human Augmentation
- Application Examples:
  - Aircraft/Component Strip & Paint
  - Surface Finishing
  - Deriveting/Cutting
  - NDI
- Enabling Tech: Telerobotics

Challenges

- Technology Transfer
- Standards
- Workspace Sharing
- Robust Input Devices
- Cooperation

Robotics and Telepresence Research Challenges - A Department of Energy Perspective

Joseph N. Herndon
Oak Ridge National Laboratory
(615) 576-0119

Discussion Outline ...

- Overview of US DOE Application Drivers
- Robotics and Telepresence Research Challenges

The US DOE Has Many Application Drivers for Robotics and Telepresence R&D

- **Environmental restoration and waste management**
  - storage tank and buried waste site remediation
  - decontamination & decommissioning of unused faciliaties
  - waste facility and processing operations
  - analytical process automation

- **Continued operations and upgrades to existing remote facilities**

The US DOE Has Many Application Drivers for Robotics and Telepresence R&D (cont.)

- **Developments for planned new facilities**
  - fusion TPX and ITER experiments
  - Superconducting Super-Collider
  - Advanced Neutron Source
  - etc

- **Ongoing basic energy research (ie CESAR)**

- **Improving US economic competitiveness through transfer of engineering and manufacturing technologies to US industry**
  - on-the-shelf now
  - cheap to implement

Robotics and Telepresence Research Challenges

1. Modular, reliable manipulation and mobility systems
   - quickly replace failed hardware in the hazardous environment
   - easy repair of failed modules of system
   - easier upgrades for system improvements
   - reconfiguration allows for more reusability of hardware across applications

Robotics and Telepresence Research Challenges

2. Improved, cost-effective control systems
   - general reusable control architectures
   - modular reusable software
   - eliminate "home cooking" to reduce costs
   - software for automatic generation of algorithms
   - push use of robust and simple intelligent control approaches (fuzzy logic, etc)

Robotics and Telepresence Research Challenges

3. Improved human-machine interfaces
   - generic human-machine interfaces
   - generalized master control for telerobotic systems
   - effective sensor-based operator assists for selective automation and collision detection
   - display of data in useable and concise formats
   - impact of virtual reality approaches unclear

Robotics and Telepresence Research Challenges

4. MOST IMPORTANTLY: Cost-effective evolution of systems from mainly laboratory environments to application environments
   - environmental and radiation hardening
   - constant attention to reliability, maintainability, and cost, a particularly difficult challenge for researchers
   - concerted attention to the machine-environment interfaces

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES

CHARLES R. PRICE

ROBOTICS SYSTEMS TECHNOLOGY BRANCH

SOAR '93

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES	AUTOMATION AND ROBOTICS DIVISION	
	CHARLES R. PRICE	AUGUST 4, 1993

ISSUE # 1

PROGRAM MANAGERS THINK THAT

SPACE ROBOTS DO TOO LITTLE AND

COST TOO MUCH.

SOAR '93

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES	AUTOMATION AND ROBOTICS DIVISION	
	CHARLES R. PRICE	AUGUST 4, 1993

ISSUE #2

TECHNOLOGICAL REALITY IS THAT

SPACE ROBOTS DO TOO LITTLE AND

COST TOO MUCH.

SOAR '93

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES	AUTOMATION AND ROBOTICS DIVISION	
	CHARLES R. PRICE	AUGUST 4, 1993

"TODAY'S" SPACE ROBOT:

- CANNOT REACH INTO CONSTRAINED SPACES
- REQUIRES SPECIAL HANDLES ON WORKPIECES
- HAS TO BE CARRIED TO ITS WORKSITE
- IS HEAVY AND POWER HUNGRY
- REQUIRES EXTENSIVE OPERATOR TRAINING
- REQUIRES CONSTANT OPERATOR ATTENTION
- IS SLOW
- IS NOT FAIL OPERATIONAL

SOAR '93

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES	AUTOMATION AND ROBOTICS DIVISION	
	CHARLES R. PRICE	AUGUST 4, 1993

WHAT NEEDS TO BE INCREASED:

- DEXTERITY
- PACKAGING DENSITY
- STRENGTH / WEIGHT
- PORTABILITY
- RELIABILITY
- STANDARDIZATION
- INTELLIGENCE
  - VISION
  - PLANNING
  - CONTROL
- ROBUSTNESS
- SPEED

SOAR '93

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES	AUTOMATION AND ROBOTICS DIVISION	
	CHARLES R. PRICE	AUGUST 4, 1993

SPACE ROBOTICS CHALLENGES --

A FIVE ITEM SUMMARY:

- TRANSPORTABILITY
- GENUINE DEXTERITY
- ROBUST INTELLIGENCE
- OPERATIONAL EFFICIENCY
- CREATIVELY COST-LIMITED

SOAR '93

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES	AUTOMATION AND ROBOTICS DIVISION	
	CHARLES R. PRICE	AUGUST 4, 1993

WHAT NEEDS TO BE DECREASED:

- WEIGHT
- POWER CONSUMPTION
- VOLUME
- LABOR INTENSITY REQUIRED OF OPERATOR
- ROBOT / WORKPIECE INTERFACE OVERHEAD
- DEVELOPMENT SCHEDULE
- COST

SOAR '93

SPACE ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES	AUTOMATION AND ROBOTICS DIVISION	
	CHARLES R. PRICE	AUGUST 4, 1993

THE BOTTOM LINE:

BUILD IT...

AND THEY WILL COME.

Session R3: ROBOTICS AND TELEPRESENCE RESEARCH CHALLENGES: PANEL DISCUSSION

Session Chair: Capt. Paul Whalen

Panel Discussion on Robotics and Telepresence (R&T) Technology Challenges

Paul V. Whalen, Capt, USAF
AL/CFBA, Building 441
2610 Seventh Street
Wright-Patterson AFB OH 45433-7901
(513) 255-3671
e-mail: p.whalen@ieee.org

4 August 1993

Abstract

A two-session panel discussion was held at Space Technology Interdependency Group (STIG) Operations Applications and Research (SOAR) 93 to identify the key R&T technology challenges that various members of the STIG Operations Committee (SOC) thought were most important to their applications. Representatives of the National Aeronautics and Space Administration (NASA), US Army (USA), US Air Force (USAF), and Department of Energy (DOE) participated (see Table 1.). Panelists each presented a list of R&T technology challenges in the first session and an open-forum discussion was held in the second session. In addition to the open discussion of the second session, the items among the lists given by the panelists were compared and contrasted. The purpose of this paper is not to discuss in detail the topics that surfaced during the panel sessions, but rather to capture the essence of the discussion and its topics for archival purposes. Interested readers are encouraged to contact either the panelists or the session moderator for further discussion of the topics enumerated in the present work.

Objective of Panel Sessions

Among the explicit goals of the SOC which sponsors the SOAR are to encourage interdependent programs and to identify critical voids in technology programs. Consequently, the objectives of these panel sessions were to (1) identify the shortfalls of R&T technology that are of greatest concern to the various government agencies represented on the panel and, (2) enumerate areas of common interest that may be targets for increased interdependent research.

Format of Panel Sessions

The first session consisted of five presentations lasting 15 minutes each. Each of the panelists listed in Table 1 had a turn to present a list of three to five challenges for the R&T research community and briefly justify them.

Table 1: List of Panel Members and their credentials

Name and Mailing Address	Credentials
Mr. Joseph N. Herndon Oak Ridge National Lab (ORNL) P.O. Box 2008 Oak Ridge TN 37831-6304	Acting Division Director of the Robotics and Process Systems Division of ORNL. US DOE Task Leader for Remote Handling on the International Thermonuclear Experimental Reactor Project. Vice Chairman of the Robotics and Remote Systems Division of the ANS.
Maj Michael B. Leahy Jr, PhD SA-ALC/TIEST Bldg 183 450 Quentin Roosevelt Rd Kelly AFB TX 78241-6416	Chief of Advanced Process Technology Section of the Technology and Industrial Support Directorate of the San Antonio Air Logistics Center and Program Manager for the Air Force Materiel Command (AFMC) RACE.
Mr. Charles R. Price NASA Johnson Space Center (JSC) ER4 Houston TX 77058	Chief, Robotic Systems Technology Branch at JSC. Oversees many projects including the Manipulator Development Facility, Automated Maintenance for Space Station, and the Dexterous Anthropomorphic Robotic Testbed at JSC.
Mr. Charles M. Shoemaker US Army Human Engineering Lab Attn: ACAP Aberdeen Prov. Gnd. MD 21005	Chief, Robotics Focus Program Office at the Army Research Laboratory (ARL). Directs near-term technology base program for Office of the Secretary of Defense's Robotics Program. Managed DEMO I Unmanned Ground Vehicle program for Army.
Dr. Charles R. Weisbin NASA Jet Propulsion Laboratory (JPL) Mail Stop 196-219 4800 Oak Grove Dr Pasadena CA 91109-8099	JPL Program Manager for Rover and Telerobotic Technologies and Senior Member of the Technical Staff. Co-chairman of the R&T Subcommittee of the SOC and the NASA Telerobotics Intercenter Working Group.
Capt Paul V. Whalen (Moderator) AL/CFBA, Building 441 2610 Seventh St WPAFB OH 45433-7901	Program Manager for the Human Sensory Feedback (HSF) for Telepresence program at the Armstrong Lab. Member of the R&T Subcommittee of the SOC and one of the principal organizers of the R&T sessions of the SOAR Symposium.

The second session was an open discussion among the panelists, the audience, and the session moderator. During this session, panelists had the opportunity to advocate their list of challenges in view of those from the other panelists and further detail issues presented in the first session.

Overview of Session 1 Presentations

Copies of the viewgraphs for the five presentations are included in these SOAR Proceedings. Brief comments on each of the presentations follow.

DOE

The DOE was represented by Mr. Joe Herndon of ORNL. Most of the ORNL R&T technology is driven by environmental restoration and waste management efforts. The DOE has been working on cleaning up hazardous waste storage tanks and buried waste sites for some time. Since the condition of the containers is typically poor and the inventory data sparse, teleoperated manipulator systems must be used to extract the waste containers for repackaging. In addition, unused facilities which have been contaminated by radioactive materials must be decontaminated and decommissioned. These initiatives alone are significant applications for the R&T technology for DOE, but they are also pressed to make plans for new facilities such as the super-conducting super-collider (SSC)[1] and emphasize technology transition to industry.

The R&T challenges listed by ORNL were:

- Modular, reliable manipulation and mobility systems
- Improved, cost-effective control systems
- Improved human-machine interfaces
- Cost-effective evolution of systems from laboratory to application environments

USAF

The USAF was represented by Major Michael B. Leahy Jr. of the San Antonio Air Logistics Center (SA-ALC) Robotics and Automation Center of Excellence (RACE). The RACE is required to work in a depot maintenance environment. This is a cost-driven environment which demands *judicious* technology insertion rather than trying to use anything that is hot out of the laboratory. The processes and tasks that are targeted by the RACE are generically called Air Logistics Center (ALC) operations. Many of the tasks that must be performed in ALC operations are very low-volume, manpower intensive tasks. A typical task may consist of removing rivets from a damaged section of aircraft skin, cutting it out, cutting a new piece of skin to match the shape of the old piece, deburring the new skin, and re-riveting it in place. The RACE is looking towards telerobotics to achieve a higher degree of productivity and process improvement rather than just a higher degree of automation. They seek to augment humans rather than trying to replace them. However, to do this means that the telerobotic tools must be easier to use than the existing tools or the workmen will not adopt the new systems. This, of course, drives home the need for reliable systems with top-notch human-machine interface for ease of operation.

The R&T challenges listed by RACE were:

- Transfer of existing component technologies to commercial sector
- Community-wide standards for hardware and software
- Safe, reliable methods of allowing shop floor personnel to share workspace with robotic systems
- Robust input devices for operator-friendly user interface
- Cooperation among researchers at all levels in Department of Defense (DOD), national labs, NASA, and universities.

[1] At the time of this writing, funding for the SSC is under Congressional scrutiny. By the time these proceedings are published, a decision should have been made about continuing support for the SSC.

NASA JSC

JSC identified the Achilles' heel of space robotics: robots, in fact, do too little for mission success and cost too much. To make matters worse, program managers are aware of this reality. Some of the limitations of current space robots that were cited included poor workspace due to oversized limbs, lack of self mobility, large weight and power consumption, extensive operator training, need for continual monitoring, and lack of fault tolerance. These observations led to a list of items which need to be increased. That list included dexterity, packaging density, strength-to-weight ratio, portability, reliability, standardization, intelligence, robustness, and speed. The items needing reduction were weight, power consumption, volume, operator intensity, robot/workpiece interface overhead, development time, and cost.

The R&T challenges listed by JSC were:

- Transportability (ground to orbit or ground to lunar)
- Genuine dexterity (manipulator dexterity equivalent to astronaut in space suit)
- Robust intelligence (integrated systems with fault tolerance)
- Operational efficiency (shorter training and less support required)
- Creatively cost-limiting development (need fresh ideas on design)

USA.

The USA was represented by Mr. Charles Shoemaker of the ARL. The ARL is primarily concerned with Unmanned Ground Vehicles (UGVs). Although they strive towards autonomous vehicles, their current thrust is teleoperated ground vehicles. Through the use of supervisory control of UGVs, they plan to make optimal use of a reduced manpower pool. In addition to the difficult technology challenges of complete autonomy, the acceptance of autonomous systems by operational users (field commanders) is generally not very high. This is due, in part, to poor demonstrated reliability of current systems and their lack of versatility. The ARL is currently retrofitting fielded combat vehicles, such as the High Mobility Multi-Wheeled Vehicle (HMMWV), with optional robotic functionality while maintaining its ability to be operated manually. This kind of system is far more acceptable to field commanders because it has back-up functionality and can be easily mobilized with other unmodified vehicles.

The R&T challenges listed by ARL were:

- Supervisory control of UGVs
  - On-board autonomy for mission function and mobility
  - Data compression and reconstitution
  - Reconfigurable man-machine interfaces
- Optional robotic functionality for manned systems
  - Non-intrusive actuation and control packages
  - Minimum volume, low-power consumption systems

  - Rugged, reliable, and maintainable systems
  - Capability for quick disconnect or back-drivable
  - Built-in diagnostic functions

- Innovative mobility platform technology
  - Stability
  - Recovery from rollover
  - Low power consumption

NASA JPL

Much of the research activity described by the JPL centered on mobility for planetary exploration and on-orbit robotic system teleoperation. Plans for a Mars rover which meets stringent weight, power consumption, and heat dissipation requirements appear to be the primary driver for the planetary rover research. The Mars rover must be extremely robust to environmental extremes (such as temperature, wind, etc.), and able to navigate in an unstructured (mostly unknown) environment with very sparse interaction from earth due to the communication delays. These requirements dictate conflicting requirements on the level of autonomy for the rover system. To cope with the difficult navigation requirements, it needs a powerful computing system with sophisticated reasoning algorithms. However, the low power, low weight, and environmentally hardened specifications eliminate all but the most primitive microprocessors because it must be a space qualified microprocessor. This, indeed, generates some difficult technology challenges which are listed below.

- Realtime perception and goal identification with limited computing power
- Ability to navigate with sparse terrain mapping data
- Need for systematic benchmark experiments to compare systems
- Increased fault tolerance and error recovery capability
- Ability to navigate autonomously when out of visual range from the lander platform

In addition to the rover research, the JPL is working to develop improved telerobotic systems for space and terrestrial operations. They have work underway in manipulator modelling and control, real-time planning and monitoring, navigation in outdoor terrain, real-time sensing and perception, human-machine interface and overall system architectures [2]. The R&T technology challenges cited by the JPL for space robotics were:

- Automated operation of remote dexterous robots from the ground
- Compilation and concatenation of robots' skills into publicly available libraries of motion primitives
- Need for instrumented end-effectors with improved dexterity
- Methods of determining object verification and pose refinement with limited computing resources
- Need for sensory skins for obstacle avoidance
- Methods for safe and robust control of manipulator/environment interaction

Overview of Session 2 Discussion

The moderator opened the second session by enumerating observations about commonalities between the various panel presentations in the first session. The list of items and the organizations that shared them included:

- Rover and mobility concerns (ARL, JPL, JSC)
- System concerns
  - Low-power, light-weight (ARL, JPL, JSC)
  - Modularity and reconfigurability (DOE, JSC, ARL, RACE)
  - Reuseable code and control architectures (DOE, RACE)
  - Standardization and metrics (DOE, RACE, JSC, JPL, ARL)
  - Reduced cost (DOE, JSC, RACE)
  - Low-bandwidth communication and control (ARL, JPL)
  - Improved end-effector dexterity (JPL, JSC, DOE)
  - Generic telerobotic (man-machine) interface (DOE, RACE, ARL)

Cultural Acceptance of R&T and Autonomy

The open discussion began with panelists voicing concern about the social acceptance of autonomy among the user community. The lack of faith in autonomous robotic solutions has hampered several attempts to field systems. For instance, ARL has been unable to gain any interest among its field commanders for autonomous vehicles that could be used for reconnaissance or targeting. Instead, the ARL has chosen the strategy of retrofitting already-accepted vehicles with *optional* teleoperated capabilities. Acceptance for such systems has been far greater than for specialized autonomous solutions. Using this strategy allows them to gradually introduce autonomy in the systems as the technology becomes proven.

RACE advocated semi-autonomous systems as a bridge between what the user community wants and what the research community wants to provide. The users want something simple, cheap, easy to operate, and reliable that will help improve their processes. The researchers, on the other hand, typically want to provide high-technology solutions that do not have proven reliability. Implementing semi-autonomous systems makes use of existing technology that has proven reliability but also allows new technology to grow *in* the application as it is proven. Thus, the autonomous function toolbox gains tools to draw upon as the technology develops. This tends to move the overall system farther from the manual teleoperation end of the spectrum and closer to the purely autonomous robot end as time goes on.

Along with the construction and manning of the proposed space station, the space community has a growing need for increased autonomy. As the number of missions and on-orbit hours increase over the years, space operations become more production oriented and less unique. Maintenance of space platforms, such as the space station, will require many routine operations that will necessarily be automated because of the time involved in doing them. The Flight Telerobotic Servicer (FTS) program was to design a fully autonomous vehicle for maintenance operations on the space station. After spending over $200M the program was cancelled before it could reach flight demonstration because of cost overruns and technical problems. This was a jolting reminder that space robotics is still technically in its infancy and appropriate "baby" steps should be taken before another

overly ambitious project will receive support from NASA. The lessons learned from the FTS will likely not be forgotten soon.

Role of virtual reality (VR) in R&T

The role of VR in R&T was the next topic of discussion. There are obvious overlaps between technologies developed for VR and those developed for R&T. Several of its more obvious roles were identified. Examples were off-line simulation and training. In general, panelists agreed that realtime VR was still a tough challenge because of the computational burden and the bandwidth limitations imposed by the amount of data that must be communicated to the user.

Although the visual display is an integral part of both VR and R&T, the unique facet of R&T that has yet to be adequately addressed by the VR community is force and tactile feedback. There is a common tendency to focus one's attention on visual display when discussing VR systems. For a VR system to achieve full *immersion* of the operator, it must also have audio, force and tactile feedback. There is a widely recognized technology void in the area of developing force-reflecting exoskeleton systems for the whole arm as well as for the fingers of the hand. The fundamental limitation in design of force-reflecting exoskeletons is the lack of suitable actuator technology. The combined requirements for small size, light weight, high power density, and high actuation bandwidth leave virtually no actuator technology candidates standing. In the view of the author, this is perhaps the most serious limitation of future VR and R&T system development.

Importance of Force-feedback

The importance of force-feedback became the next discussion topic. There were proponents of force-feedback who argued that it has been proven to increase teleoperator system performance in many tasks as demonstrated by the DOE and others. There were also people who stated unequivocally that their tasks did not benefit from the addition of force-feedback to the telerobotic system. One example of such an application is the teleoperation of heavy equipment for Rapid Runway Repair (RRR). In this case, a full-scale backhoe is teleoperated to excavate unexploded ordnance and repair craters in runways damaged by air attack. The Air Force Construction Robotics Program at Tyndall AFB FL (HQ AFCESA/RA) has evaluated force-feedback for this task and found that it is not beneficial. This is not surprising when one considers that a backhoe operator does not use force-feedback information even when manually operating his equipment. However, the benefit from force-feedback for other tasks is undeniable. For instance, part mating is inherently a force-domain task and providing force-feedback information to the operator has improved task performance in several studies (for example, see [1].) .

Customer Involvement

Panelists agreed that the research community in R&T, like that of many other technologies, has not been very good at understanding and addressing the constraints of their technology using customers. To be effective, researchers must recognize the constraints of their users and make serious attempts to work within them. Typical constraints may be size limitations, weight limitations, cost limitations, reliability requirements, etc. Some constraints are even time based such as deadlines for delivery. There are other options for most mission requirements and R&T solutions will not be welcome until they are competitive with the other options.

Need for Standards and Metrics among R&T Community

Cost, development time, and reliability are perhaps the weakest points for developing R&T solutions. All of these factors could be improved with accepted standards which would boost the commercialization of technology. Currently there are no commercial systems that allow systematic interface of various sensors into robotic systems. The R&T community needs to work towards standards that will allow researchers and system developers to pull component systems off the shelf and use them without the extensive integration work that is currently required. The idea of establishing standards for the whole field of R&T is overwhelming and, even if it were possible, it would probably stifle some areas of development. On the other hand, a "bottom-up" approach to establishing standards could benefit all parties. Well-formulated standards for component systems can be aggregated over time into more pervasive standards as they mature.

Metrics are also needed to make meaningful comparisons between similar solutions to the same problem. For instance, a mobility metric would be useful to compare unmanned ground vehicles that use completely different modes of mobility (e.g., legged, wheeled, tracked, etc.). Even within a single mode of mobility, there is currently no agreed-upon metric by which comparisons can be made. Although grey areas of comparison will always persist, a good metric could at least help identify the very good and very bad solutions.

Collision Detection and Avoidance

A brief discussion on collision detection and avoidance concluded that viable solutions are near maturity. The JPL is concluding a study on range sensors this year and will be using that information in its development of skin-type contact sensors. Most of the panel members said they would probably use collision detection and avoidance technology, but they were not actively pursuing it. The army mentioned that the type of collision detection they are interested in is the same kind that the Department of Transportation (DOT) is working on for the Intelligent Vehicle Highway System (IVHS). The IVHS is envisioned to eventually have autonomous vehicles shuttling people between destinations with little or no operator involvement. Avoiding collisions in emergency situations and maintaining safe spacing between vehicles on the highway are tasks that will require sophisticated collision detection and avoidance capability.

Conclusions

The two sessions were intended to identify important technology areas that the various member agencies of the SOC may have in common. There were several areas that were immediately obvious after the first of the two sessions which are listed herein. There are undoubtedly others that are common but are of lesser importance to the individual agencies as represented by the selected panelists. Having identified some common areas of interest, opportunities have been identified for increased interaction and interdependency among the participating agencies at various levels. This interaction may lead to reduced duplication and/or joint funding for specific programs in the future. This, of course, is the primary purpose of the SOC which sponsors the SOAR. It is this author's hope that these two panel discussion sessions have furthered that cause.

References

[1] B. Hannaford, L. Wood, D. McAffee, and H. Zak. Performance Evaluation of a Six-axis Generalized Force-Reflecting Teleoperator. *IEEE Trans. on Systems, Man, and Cybernetics*, 21(3):620–633, 1991.

[2] C. Weisbin and D. Perillard. Jet Propulsion Laboratory Robotic Facilities and Associated Research. *Robotica*, 9:7–21, 1991.

Session R4: REMOTE INTERACTION WITH SYNETHETIC ENVIRONMENTS

Session Chair: Dr. Harold Hawkins

48.
C-3.

N94-34024

Shared Virtual Environments for Aerospace Training

R. Bowen Loftin
University of Houston–Downtown
Houston, TX

Mark Voss
LinCom Corporation
Houston, TX

Virtual environments have the potential to significantly enhance the training of NASA astronauts and ground-based personnel for a variety of activities. A critical requirement is the need to share virtual environments, in real or near real time, between remote sites. It has been hypothesized that the training of international astronaut crews could be done more cheaply and effectively by utilizing such shared virtual environments in the early stages of mission preparation. The Software Technology Branch at NASA's Johnson Space Center has developed the capability for multiple users to simultaneously share the same virtual environment. Each user generates the graphics needed to create the virtual environment. All changes of object position and state are communicated to all users so that each virtual environment maintains its "currency." Examples of these shared environments will be discussed and plans for the utilization of the Department of Defense's Distributed Interactive Simulation (DIS) protocols for shared virtual environments will be presented. Finally, the impact of this technology on training and education in general will be explored.

N94- 34025

Surgery Applications of Virtual Reality

Dr. Joseph Rosen
Dartmouth-Hitchcock Medical Center
Lebanon, NH

Virtual reality is a computer-generated technology which allows information to be displayed in a simulated, bus lifelike, environment. In this simulated "world", users can move and interact as if they were actually a part of that world. This new technology will be useful in many different fields, including the field of surgery. Virtual reality systems can be used to teach surgical anatomy, diagnose surgical problems, plan operations, simulate and perform surgical procedures (telesurgery), and predict the outcomes of surgery. The authors of this paper describe the basic components of a virtual reality surgical system. These components include: the virtual world, the virtual tools, the anatomical model, the software platform, the host computer, the interface, and the head-coupled display. In the chapter they also review the progress towards using virtual reality for surgical training, planning, telesurgery, and predicting outcomes. Finally, the authors present a training system being developed for the practice of new procedures in abdominal surgery.

N94- 34026

A STUDY OF NAVIGATION IN VIRTUAL SPACE

Rudy Darken & John L. Sibert
The George Washington University
Department of Electrical Engineering and Computer Science
801 22nd St. NW
Washington, D.C. 20052
darkenlsibert@seas.gwu.edu

Randy Shumaker
The U.S. Naval Research Laboratory
Information Technology Division
Code 5500, 4555 Overlook Ave. SW
Washington, D.C. 20375
shumaker@itd.nrl.navy.mil

ABSTRACT

In the physical world, man has developed efficient methods for navigation and orientation. These methods are dependent on the high-fidelity stimuli presented by the environment. When placed in a virtual world which cannot offer stimuli of the same quality due to computing constraints and immature technology, tasks requiring the maintenance of position and orientation knowledge become laborious. In this paper, we present a representative set of techniques based on principles of navigation derived from real world analogs including human and avian navigation behavior and cartography. A preliminary classification of virtual worlds is presented based on the size of the world, the density of objects in the world, and the level of activity taking place in the world. We also summarize an informal study we performed to determine how the tools influenced the subjects' navigation strategies and behavior. We conclude that principles extracted from real world navigation aids such as maps can be seen to apply in virtual environments.

INTRODUCTION

Orientation and navigation are fundamental components of movement in any space. This is particularly true in virtual spaces where tasks involving movement of any kind become difficult due to the low-fidelity stimuli presented by the virtual environment. Our focus in this exploratory research has been on navigation tasks and human behaviors associated with these tasks in differing worlds with various cues and tools. The approach taken begins with a classification of virtual worlds based on their spatial attributes and an enumeration of navigation tasks performed in these worlds. Considering human abilities, both innate and artificially enhanced, we have built a set of tools designed to aid in performance of navigation tasks. Results of an informal empirical study are presented suggesting that a relationship exists between cues and tools available in an environment and navigational behaviors exhibited by the user.

A PRELIMINARY CLASSIFICATION OF VIRTUAL SPACES

We have chosen to classify virtual worlds based on three attributes: size, density, and activity. We do not claim that this classification is precise or complete. A complete classification scheme could in fact be a useful metric for the evaluation of virtual worlds and interaction techniques associated with them.

Size

A small world is described as any world in which the entire world can be viewed in detail from a single vantage point. Small worlds tend to focus the user's attention on a single object or group of related objects. An example of such a world is the virtual windtunnel (Bryson & Levit, 1991; Bryson & Gerald-Yamasaki, 1992).

A large world is defined by Kuipers and Levitt (1988) as a "space whose structure is at a significantly larger scale than the observations available at an instant." We modify this, making it more geometric, by stating: there is no vantage point from which the entire world can be seen in detail. This keeps us consistent with our definition of a small world. A large world may or may not be of finite size. An infinite world is defined as one in which we can travel along a dimension forever without encountering the "edge of the world."

Density

A sparse world has large open spaces in which there are few objects or cues to help in navigation. An example of this is a naval simulation which is populated by only a few objects of interest. Experience has shown that subjects in such a space easily become disoriented (Darken & Bergen, 1992). Contrarily, a dense world is characterized by a relatively large number of objects and cues in the space. An example of this would be the simulation of an urban area with many closely spaced buildings.

Another aspect of density is the distribution of objects in the space. As the distribution approaches uniformity, the positions of objects become much more predictable. On the other hand, if objects are found clustered around a relatively small number of locations, a space with a relative number of objects sufficient to be dense can actually be sparse.

Activity

The level of activity of objects within a world can be static or dynamic. In a static world, the positions of objects do not change over time. This represents the simple end of the activity scale. Dynamic worlds are worlds in which objects move about, thereby increasing the complexity of the navigational task. This movement can be deterministic or nondeterministic in nature. Worlds can be characterized along a continuum from fully determined, where all of the objects move deterministically, to fully nondetermined, where all objects move randomly.

NAVIGATION

We use the term "navigation" to describe any process of determining a path to be traveled by any object through any environment. For this study, that object is always the user's viewpoint in the virtual world. The ideas and tools for navigation presented here have been developed for application to the real world, or at least adapted for application to virtual worlds with similar dimensionality and properties to the real world. However, virtual environment technology enables the ability to create environments where we radically alter physical scale, time scale, sensor modality (e.g. feeling electromagnetic forces, seeing sound, hearing texture, etc.) and sensor sensitivity. This provides the potential to consider creating entirely synthetic environments that map various phenomenon of interest into modalities to permit "direct" sensory exploration of phenomenon. This capability may become valuable in the "visualizing" and understanding of otherwise difficult to understand abstract features and interactions. Many of the concepts, and even some of the actual tools of real world navigation are directly applicable to virtual worlds representing both possible and entirely synthetic phenomena.

Human Navigation

Humans are thought to form cognitive maps of their environments for use in navigation (Stevens & Coupe, 1978; Howard & Kerst, 1981; Goldin & Thorndyke, 1982). These maps encode spatial information such as landmarks and distances. It is believed that avian cognitive maps utilize a sophisticated multisensory landmarking technique in which no distinction is made between visual, acoustic, or olfactory landmarks (Baker, 1984). Also, the ability to fly greatly alters the cognitive map's range, detail, and complexity. Lynch (1960, 1965, 1959, 1958) developed a set of generic components which he hypothesized are used to construct cognitive maps of urban environments. They include:

- *Paths*: linear separators, examples include walkways and passages.
- *Edges*: linear separators, such as walls or fences.
- *Landmarks*: objects which are in sharp contrast to their immediate surroundings, such as a church spire.
- *Nodes*: sections of the environment with similar characteristics. For example, a group of streets with the same type of light posts.
- *Districts*: Logically and physically distinct sections. In Washington, D.C., they might be Foggy Bottom, Capitol Hill, etc.

Through the ages, humans have developed techniques for navigation and piloting to compensate for their perceptual system's limited ability to effectively utilize the physical cues available in nature. The primitive technique of dead reckoning is used today as a simple yet effective navigation method. The navigator marks the present position and orientation. This information is used, along with the distance traveled in a straight line, to determine a future position (Bowditch, 1966). Trailblazing is performed in a similar fashion. Typically, physical markers are left behind to encode past positions or information concerning those positions for future retrieval. A more modern tool is the global position indicator which utilizes two satellite signals to accurately determine latitude and longitude. This information can be used with a local map for accurate navigation.

One of the most effective tools for navigation is, of course, the map. Physical map organization and display and the relationship between the physical map and its associated cognitive map are also at issue. Boff and Lincoln (1988) present three fundamental design principles for maps:

- The two-point theorem states that a map reader must be able to relate two points on the map to the corresponding two points in the environment. This will orient the space properly to facilitate the map's use for navigation.
- The alignment principle states that the map should be aligned with the terrain. That is, a line between any two points in space should be parallel to the line between those two points on the map.
- The forward-up equivalence principle. The upward direction on a map always shows what is in front of the viewer.

In addition to traditional maps, Simutis and Barsam (1980) describe the use of contour maps for navigation and orientation. The terrain contour itself is used as a cue to maintain direction.

An Informal Study of Navigation

For our initial study, we chose a virtual environment that is both simple and relatively similar to a physical environment. The world consists of a large rectangular plane which can be randomly filled with a varying number of typical objects.* We also focused on three different navigation tasks: *exploration*, where the primary goal is gaining familiarity with the environment; *naive search*, where the subject is searching for an object when its appearance but not its location, is known; and *informed search*, when the subject has some knowledge about the location of the object.

*. We used ships since the closest physical analog is a large tract of open sea.

The study included nine subjects, seven male and two female[†] all of whom have a technical background and are experienced computer users. Only three of the subjects had any experience using the apparatus and none had any previous knowledge of the subject matter of the study. A Fake Space Labs, Inc. BOOM2C display was used for high resolution, monochromatic display and mechanical tracking. The Audio Cube by Visual Synthesis Inc.[‡] was used for spatial audio.

For each trial, a large world was randomly configured based on the number of objects required (sparse or dense world) and the tools to be made available. The initial viewpoint location was marked with a flat square on the ground plane and the target was placed randomly at some minimal distance from the initial viewpoint location. The ground plane was represented as a square grid. The objects were identical ships. The target was a small pyramid. One button on the BOOM2C was used for forward movement in the view direction and the other for backward movement. Movement speed was not variable and movement through the ground plane was not allowed. Due to the use of primarily distant viewing, stereoscopy was not utilized.

Before their initial participation, subjects were informed as to the nature of the study and what they would be seeing in the worlds. Before each treatment, subjects were given information about the structure or representation of the tool(s) to be used but were never prompted with suggested strategies. For example, the components of the mapview and the orientation of the coordinate systems were described but subjects were not told how to use the tools. The task was described as having three primary parts:

1. Move through the space at will trying to view as much space as possible.
2. Search for the target object.
3. On cue, return to the start position.

Each subject was instructed to browse the space in an investigative fashion. Spatial knowledge gathered in this step is useful in the subsequent search tasks. At some random time before the target was visible to the subject, each was told to search for the target object. After moving sufficiently close to the target, an audible bell would sound signalling the subject to return to the initial position (marked by a square). During each trial, subjects were asked to freely describe choices being made, strategies, and general actions.

Subject behavior was recorded in written notes documenting observations made by the evaluator and comments made by the subjects during and after each trial. Of particular interest was data on positional or orientational information being gleaned from the environment or the tools and strategies used to accomplish any part of the task. Each scenario of tool(s) and world type was tried by different subjects until a generalization could be made on behavior in that scenario. Typically, five to six trials per scenario were used.

Tool Descriptions and Observations of Use

We have implemented a toolset which consists of a subset of the navigation techniques used in the physical world. Table 1 lists the techniques and, for each of them, the real world analog which we used as our guide in developing each technique.

[†] Although some studies have indicated gender variance in navigational behavior, we did not observe any gender based differences.

[‡] The Audio Cube uses a cube of eight external speakers rather than headphones to position the sound sample.

Landmark Scenario

Synthetic landmarks can be placed in the world. These landmarks are distinct from other objects in the space and are placed randomly when the environment is created. The landmarks we used were simple rectangular columns, but they were considerably larger than the ships (figure 1). Subjects began by scanning the space from the starting location. They attempted to locate easily identifiable configurations of landmarks or clusters of ships. If they were able to locate a configuration of landmarks which also provided directional information, such as an "L" shape, their homing performance was improved.

Technique	Real World Analog
flying	avian navigation
spatial audio	avian landmarking
breadcrumb markers	trailblazing
coordinate feedback	global position indicator
districting	urban environmental cues
landmarks	urban environmental cues
grid navigation	contour map orientation
mapview	map organization & presentation methodologies

Table 1: Navigation techniques in the toolset.

When subjects began moving through the space they attempted to use landmarks to separate the space into segments. If the landmarks were configured in such a way as to make it difficult to use them as separators, subjects had a tendency to become disoriented and repeatedly search the same space. During this searching phase, subjects were also trying to maintain a direction for home.

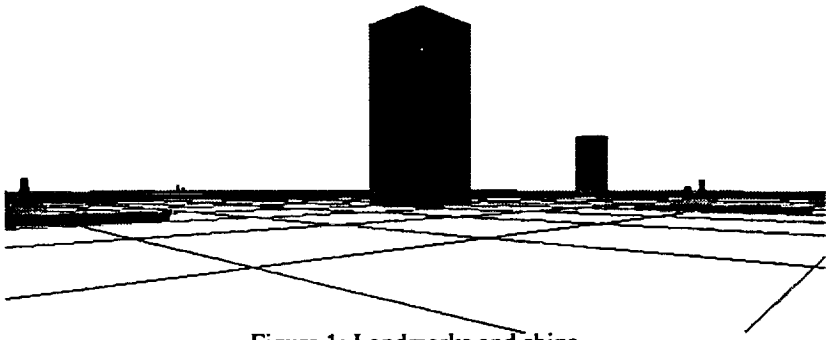


Figure 1: Landmarks and ships.

During the homing phase, all subjects initially moved in an inaccurate direction indicating that their ability to maintain an accurate home direction was poor. Furthermore, those subjects who were unable to glean any directional information from landmark configuration were forced to perform the same kind of exhaustive search to find their way home that they had performed to find the target in the first place.

When a synthetic sun was added, all subjects' performance in both phases of the search improved. The landmarks were still used to separate the search space and make the search for the target more efficient but the sun provided much better directional information. This seems to result from two characteristics of the sun; its relative immobility and its

visibility throughout the space make it an absolute directional marker. In contrast the most distinctive configurations of landmarks can only provide directional information relative to a local region.

Coordinate Tools Scenario

A coordinate feedback system displays a continuous textual readout of either Cartesian or polar coordinates of the subject's current position. This is similar to the type of information available from the global position indicator.

Subjects determined their orientation by making exploratory movements and observing how their coordinates changed. With Cartesian coordinates, the subjects tended to align their view direction with one of the axes of the world grid and move back and forth while observing changes in the coordinates. They would then turn ninety degrees and repeat the back and forth movement. With polar coordinates, subjects tended to combine small back and forth movements with sweeping from side to side.

The coordinate tools proved most useful for the homing task. Subjects were able to remember the coordinates of their starting place and quickly recognized the relationship between their current and starting positions. In both cases the subjects tended to treat homing as a separable task (Jacob & Sibert, 1992) where movement and searching were performed disjointedly. We feel that this task separation is an artifact of the tools rather than something that is inherent in the task. With the polar tool, subjects would first adjust the bearing and then the range or vice versa. With the Cartesian tool subjects treated movement in x and y separately. The Cartesian coordinate tool was also somewhat useful in the target search since it could be used easily to partition the space into quadrants.

Breadcrumbs (or Hansel and Gretel Scenario)

A system of marking the space with a visual marker (a simple unmarked cube which we call a breadcrumb) was implemented. This mechanism can be used manually, requiring the user to specify where markers should be dropped, or automatically, dropping markers at a constant frequency along the user's path. This method was originally intended to be used as a trail making mechanism but was found to be used more as a manual landmarking technique where subjects would mark positions in space with semantic information. Subjects typically would mark the start position to simplify their return later in the trial. This was done in such a way as to be directional (See Landmarks Scenario). The criteria for dropping a marker depended on the strategy being employed. If an exhaustive search was required, markers were dropped at a regular frequency in space to mark places as searched. If dead reckoning was being performed, markers were dropped along a straight line between two positions. Subjects also attempted to create a directional indicator with the markers showing a direction change if possible.

Subjects exhibited behavior similar to that in the landmark treatment. Since the markers were nondirectional, maintaining orientation was a problem. Only relative information was available from the markers. Breadcrumbs were also used in an automatic mode in which markers were dropped at some set frequency in time. This technique was useful only for leaving a trail or as a method of marking searched spaces because it was not directly in the subject's control.

Flying Scenario

When we allow flying as a means of movement, we are effectively adding the third spatial dimension as a tool if we keep the navigation task two-dimensional. This is reflected in the initial action taken by subjects, flying up to get a bird's-eye view of their surroundings. They then maintained their altitude while searching for the target. The "fly where you look" style of movement made this difficult but a relatively steady altitude could be maintained with slight up and down fluctuations. This has the effect of changing the scale at which they view the world and is somewhat analogous to using a map. A map is, after all, a small scale representation of important characteristics of a space. The

major difference is that, when flying in this way, a subject is combining map reading, navigation and movement into a unified task. A further indication that the subjects are integrating these tasks is the nature of their flight path. Subjects tended to simultaneously move the BOOM and depress a movement button yielding parabolic changes in direction. Simultaneous movement and change of direction was almost never observed in any of the other treatments.

Mapview Scenario

The mapview is a dynamic map linked to the viewpoint which can be either aligned with the world or aligned with the viewpoint. The distinction is related to the map organization and presentation methodologies previously described by Boff and Lincoln (1988). The map in our mapview tool appears to float within the lower part of the field of view so that the subject can consult it at will by glancing down, yet it does not obscure the environment when the subject is looking around. The map shows the locations of; the starting point, ships, landmarks (if present), and the subject (figure 2). The two treatments of mapview differ in their rules for orientation. In the view-aligned treatment, the map

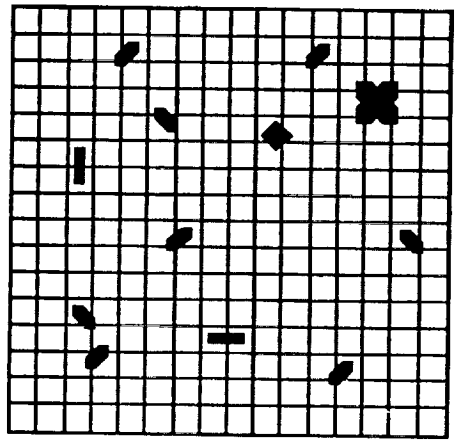


Figure 2: Schematic illustrating the map for mapview. X represents the start point and the diamond is the "you are here" marker. Other symbols represent ships; no landmarks are shown.

is always oriented with its top in the direction of the subject's view (figure 3a). This is analogous to navigating in a car with the map on your lap and its top oriented towards the dashboard regardless of the direction in which the car is moving. This behavior is characteristic of travel between cities. Our other treatment, world-aligned, keeps the map in constant alignment with the coordinate system of the world (figure 3b). This is somewhat analogous, in the car navigation example, to twisting the map so that the street you are driving along is aligned with its representation on the map. People tend to exhibit this behavior when they want to make sure they are turning in the correct direction at the next corner. Only this treatment satisfies the alignment and forward-up principles.

Because the map includes the starting point, it was unnecessary for the subjects to remember its location. Each version of mapview had both advantages and disadvantages. The view-aligned version was more useful for exhaustively searching the space. Subjects appear to have formed a more complete cognitive map of the environment since their view of the map did not vary as they moved. On the other hand, it was necessary for them to move and watch this motion reflected by the "you are here" indicator on the map in order to determine their orientation. With the world-aligned version, subjects had no difficulty determining their orientation from the map since it conforms to the alignment principle. However, maintaining world alignment causes the map to appear to rotate when the subject changes direction. This makes it harder to maintain a consistent cognitive map of the environment and hence decreases the usefulness of the map as an aid for exhaustive search.

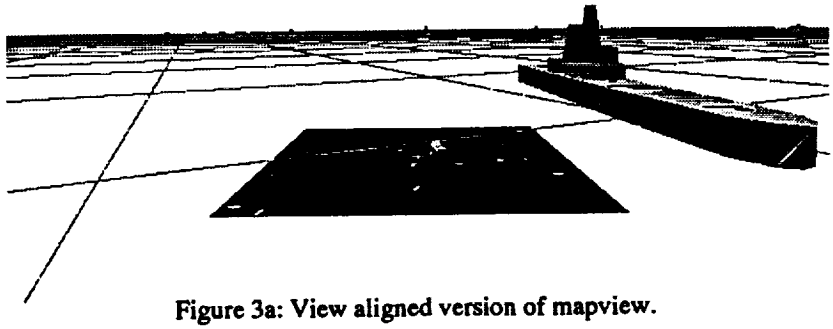


Figure 3a: View aligned version of mapview.

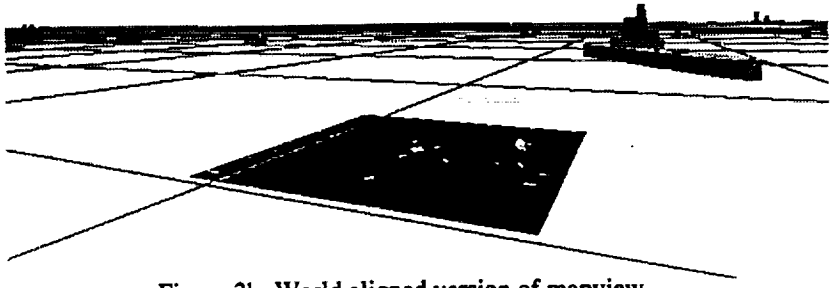


Figure 3b: World aligned version of mapview.

Other Methods

Other treatments implemented and studied include districting, spatial audio, and grid navigation. Districting was implemented as a visual subdivision of the world into four quadrants and is based on Lynch's (1960, 1965, 1959, 1958) districts described earlier (See Human Navigation). The districts allowed subjects to "chunk" spatial information necessary for learning and searching tasks into pieces. Searching was performed sequentially by district. Districts could be combined together to form an image of the world as a whole.

A spatial audio cue, a steady positional tone generated using the Audio Cube (by Visual Synthesis Inc.) is used as an acoustic landmark. This is currently our only non-visual modality. The audio signal was added to the start location as a cue for the homing task. The cue was not audible throughout the world and thus offered no information when outside its range. When it became audible, it was used for rough direction finding. The spatial audio cue had the effect of enlarging the target object.

Lastly, when no other cues were available, subjects resorted to using the ground plane grid itself as a cue. The grid cannot offer assistance in position (unless an edge is used in a finite world). The orientation information available is cognitively demanding to maintain because it is purely relative information and requires attention to the grid at all times. If the grid included contour information (Simutis & Barsam, 1980), orientation would become easier and even positional information might be available.

CONCLUSIONS

The complexity of navigation tasks in virtual environments requires special attention in the development of interaction techniques pertaining to navigation aids. Our intention has been to investigate design principles and study their

relationship to user behaviors in virtual spaces. Considering the innate use of environmental cues by humans and the principles of cognitive map formation and map design developed by cartographers and planners, we developed a toolset of navigation aids for use in virtual spaces. An informal empirical study of the tools for a small set of searching tasks supports the following general conclusions:

- People tend to take advantage of environmental cues in predictable ways. They use them to partition spaces as an aid to exhaustive search. They use them to maintain direction relations performing best when the cue is statically positioned or highly predictable in its motion and when it is visible from the entire environment.
- The tools they use have strong influences on people's behavior. Our subjects showed very different behavior when they used different tools. The variation among tool treatments was much larger than the variation among subjects.
- Because the navigation tasks were constrained to be two-dimensional and were performed on a two-dimensional surface, cartographic design principles could be extended from the real world to the virtual world. Had we included a three-dimensional task, such as a hunt for a spacecraft in an asteroid belt, we doubt that our mapview would have been of much use.

These conclusions, although far from definitive, are suggestive and encourage us to consider extending our research. We must form more specific hypotheses about how design principles relate to environmental characteristics and test them with more formal studies. We also intend to extend the research to virtual environments which have less in common with the real world. We hope that by doing this in a careful and gradual way, we will be able both to extend existing principles into new domains and to develop new principles for tool building in virtual environments.

ACKNOWLEDGMENTS

This research was supported by funding from NRL (NAVY N00014-91-K-2031). Special thanks go to Allen Duckworth, the late Donald Grady, and John Montgomery of the Tactical Electronic Warfare Division for their support of our work at the Naval Research Laboratory. We would also like to thank the Computer Graphics and User Interface Group at the George Washington University for their assistance.

REFERENCES

Baker, R.R. (1984). Bird Navigation: The Solution of a Mystery? London: Hodder and Stoughton.

Boff, K.R., & Lincoln, J.E. (1988). Engineering Data Compendium: Human Perception and Performance. Wright-Patterson AFB, Ohio.

Bowditch, N. (1966). American Practical Navigator An Epitome of Navigation. Washington: U.S. Naval Oceanographic Office.

Bryson, S. & Levit, C. (1991). The Virtual Windtunnel: An Environment for the Exploration of Three-Dimensional Unsteady Flows (Tech. Rep. RNR-92-013). Moffet Field, California: National Aeronautics and Space Administration Ames Research Center.

Bryson, S. & Gerald-Yamasaki, M. (1992). The Distributed Virtual Windtunnel (Tech. Rep. RNR-92-010). Moffet Field, California: National Aeronautics and Space Administration Ames Research Center.

Darken, R., & Bergen, D.E. (1992). A Virtual Environment System Architecture for Large Scale Simulations. Proceedings of Virtual Reality 1992. 38-58.

Goldin, S.E., & Thorndyke, P.W. (1982). Simulating Navigation for Spatial Knowledge Acquisition. Human Factors, 24(4), 457-471.

Howard, J.H. Jr., & Kerst, S.M. (1981). Memory and Perception of Cartographic Information for Familiar and Unfamiliar Environment. Human Factors, 23(4), 495-504.

Jacob, R.K.J. & Sibert L.E. (1992). The Perceptual Structure of Multidimensional Input Device Selection. Proceedings of SIGCHI 1992. 211-218.

Kuipers, B.J. & Levitt, T.S. (1988). Navigation and Mapping in Large-Scale Space. AI Magazine, 9(2), 25-43.

Lynch, K. (1960). The Image of the City. Cambridge: M.I.T. Press.

Lynch, K. (1965). The City as Environment. Scientific American. 213, 209-219.

Lynch, K., & Rivkin, M. (1959). A Walk Around the Block. Landscape. 8, 24-34.

Lynch, K., & Rodwin, L. (1958). A Theory of Urban Form. Journal of the American Institute of Planners. 24, 201-214.

Simutis, Z.M., & Barsam, H.F. (1980). Terrain Visualization and Map Reading. In Pick, H.L., & Aeredole, L. (Eds.), Spatial Orientation: Theory, Research, & Application (pp. 161-193). New York: Plenum Press.

Stevens, A. & Coupe, P. (1978). Distortions in Judged Spatial Relations. Cognitive Psychology. 10, 422-437.

N94-34027

RoboLab and Virtual Environments

Joseph C. Giarratano, Univ. of Houston Clear Lake
Lac Nguyen, Software Technology Branch (PT4)
NASA/Johnson Space Center
Houston, Texas

ABSTRACT

A useful adjunct to the manned space station would be a self-contained free-flying laboratory (RoboLab.) This laboratory would have a robot operated under telepresence from the space station or ground. Long duration experiments aboard RoboLab could be performed by astronauts or scientists using telepresence to operate equipment and perform experiments. Operating the lab by telepresence would eliminate the need for life support such as food, water and air.

The robot would be capable of motion in three dimensions, have binocular vision TV cameras, and two arms with manipulators to simulate hands. The robot would move along a two-dimensional grid and have a rotating, telescoping periscope section for extension in the third dimension. The remote operator would wear a virtual reality type headset to allow the superposition of computer displays over the real-time video of the lab. The operators would wear exoskeleton type arms to facilitate the movement of objects and equipment operation. The combination of video displays, motion, and the exoskeleton arms would provide a high degree of telepresence, especially for novice users such as scientists doing short-term experiments.

The RoboLab could be resupplied and samples removed on other space shuttle flights. A self-contained RoboLab module would be designed to fit within the cargo bay of the space shuttle. Different modules could be designed for specific applications, i.e., crystal-growing, medicine, life sciences, chemistry, etc.

This paper describes a RoboLab simulation using virtual reality (VR.) VR provides an ideal simulation of telepresence before the actual robot and laboratory modules are constructed. The easy simulation of different telepresence designs will produce a highly optimum design before construction rather than the more expensive and time consuming hardware changes afterwards.

INTRODUCTION

The RoboLab concept is that of a free-flying laboratory with a telepresence robot operated by a human from the ground or the space station. RoboLab can be considered as part of a complementary space operations triage:

- Work requiring **continuous** human presence — Space Station
- Work requiring **part-time** human presence — Space Shuttle
- Work requiring **no** human presence — RoboLab

RoboLab is complementary to the space station and space shuttle The space station is ideal to support RoboLab especially when tens or hundreds of labs are linked. Full time astronaut support would then be required to:

- Link new modules
- Reconfigure existing modules
- Enhance modules with new equipment
- Resupply equipment and raw materials
- Harvest finished products
- Repair and maintain modules
- Provide detailed, on-the-spot assessment of unusual problems

RoboLab has been implemented in a software simulation using virtual reality (VR). Simulation has proven very successful in rapid prototyping and testing the feasibility of concepts. In particular, simulation is a valuable tool before hardware is constructed since it is much cheaper to do a software simulation than construct expensive hardware, especially in space.

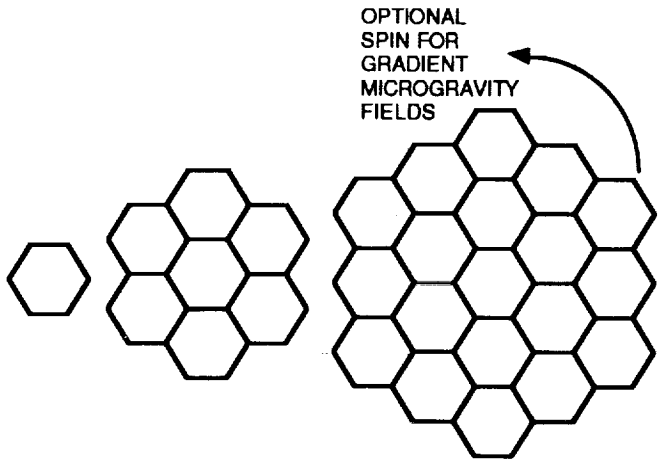
Virtual Reality is a technology that is now being applied to many fields. In common VR systems, the user wears a special helmet which is motion sensitive and provides a 3D-real-time display of a simulated scene. A special glove is worn containing sensors that are sensitive to hand motion. Other types of gloves are available which provide force feedback and other sensations so that the user can "feel" simulated objects and their characteristics such as temperature and texture.

A virtual reality simulation can be used very effectively for testing proof of concept of telepresence in space. The idea of telepresence is to allow a human operator to remotely operate a robot as if the human was present. Telepresence is very useful when the robot must operate in a hostile or dangerous environment. In space, Telepresence is also useful from an economic point of view since an astronaut's time is valued at about $40,000 an hour.

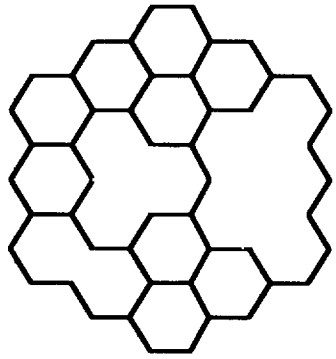
The telepresence can be performed from earth or the space station. If done on earth, no special training or background as an astronaut would be required. Ordinary scientists and engineers can use RoboLab 24 hours a day from anywhere in the world.

ROBOLAB EXTERNAL ARCHITECTURE

The RoboLab concept is to provide a low-cost, object-oriented approach to hardware development. The initial goal is to mass produce an economical, self-contained hexagonal laboratory module that can fit within the cargo bay of the space shuttle. Modules from successive flights can be linked together to produce a larger laboratory by incremental growth. Fig. 1 illustrates different configurations of RoboLab as module shells are added.



(a) INCREMENTAL ROBOLAB GROWTH



(B) ENLARGED LABS CREATED BY REMOVING SELECTED MODULE WALLS

FIG. 1
ROBOLAB GROWTH AND OPTIONS

In Fig. 1, the lab is shown in stages as a complete new outer shell is added. However, the lab is always fully operation even if a shell is not complete. Individual labs may be linked together to provide larger spaces by removing lab walls as desired. Some labs may have zero-g and larger lab spaces, while others have smaller spaces and micro gravity. Of course, if the lab spaces are made symmetric with regard to the center of mass, then these larger labs can be spun as illustrated in Fig. 1.

The main advantage of a complete shell is spinning the lab to set up a microgravity gradient. If a shell is not complete, the center of mass will not be at the center and it will be more difficult to stabilize the lab. A hexagonal shape was chosen for each lab to

facilitate incremental growth, i.e., the beehive pattern. This hexagonal shape allows easy locking of new modules and a quasi-circular shape as new shells are added. The quasi-circular, pancake shape makes it easier to spin the lab in a stable way.

Astronauts will bring new modules, link them together, enhance capabilities by replacing old equipment, perform maintenance, bring supplies, and return finished products, e.g. crystals, materials, and medical drugs, back to earth or to the space station. All modules are prewired and designed to quickly snap together. Special purpose modules may be designed for human life support.

ROBOLAB INTERNAL ARCHITECTURE

RoboLab is a facility in which operations are performed by the telepresence robots. Fig. 2 illustrates an individual module showing the robot. The goal is to provide a user-friendly telepresence system that anyone can use after minimal instruction.

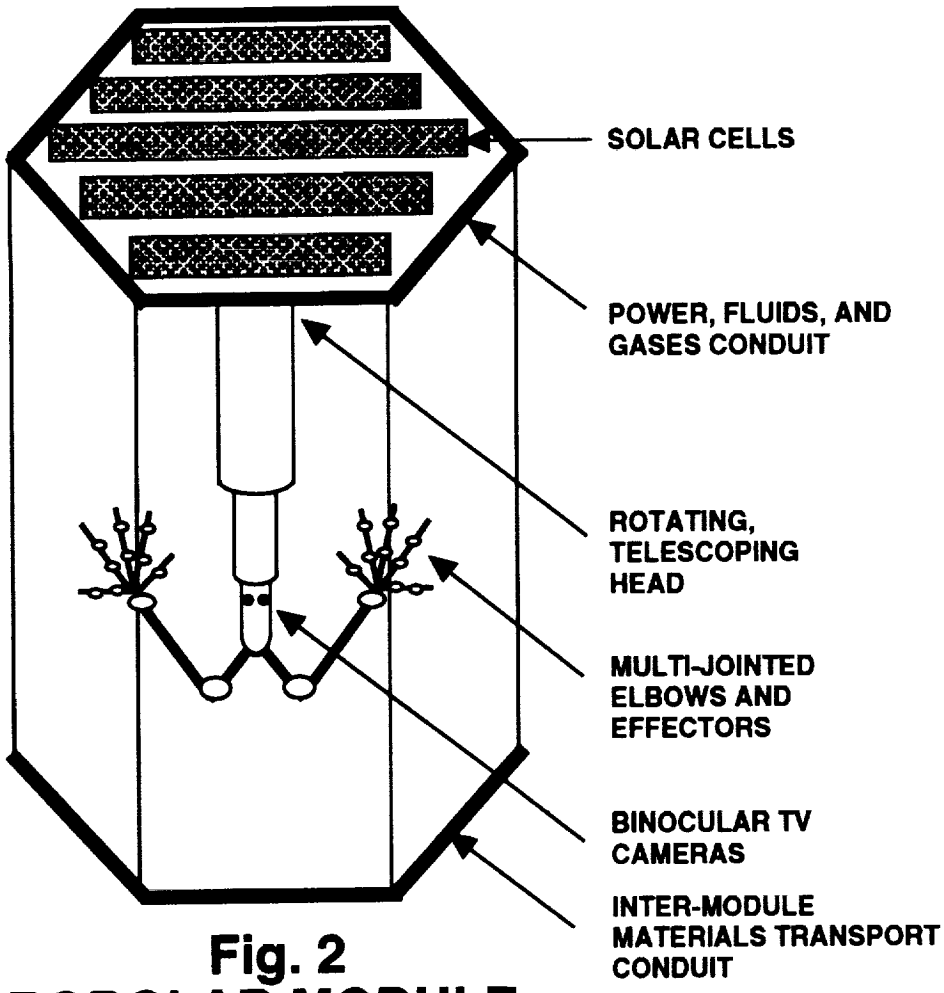


Fig. 2
ROBOLAB MODULE

The RoboLab walls are attached to a frame consisting of hollow girders containing utility conduits for power, fluids, and gases, e.g., air and water. Utilities are routed by power, fluid, and gas switches from one module to another through these ceiling utility conduits. Special modules may serve as supply depots for utilities such as fluids and gases.

As more modules are connected, the available solar power to RoboLab increases. The aggregation of this power comprises a solar power grid. Electricity from the grid may be routed on demand to those module which need more than their individual panels can provide. An active power switching system routes power from modules which need less to those which need more via the power conduits. The active power switch resides in the ceiling of each module to siphon off the required power.

The hollow girders of the floor contain an electric powered "subway" transfer system to shuttle materials from one module to another. Coffee-can sized containers can be transported on the subway train to any other module. Semi-processed materials can be transferred to other modules for finishing. Finished products can be transported to a special linear accelerator module for launch to Earth or the space station.

TELEPRESENCE ROBOT

The telepresence robot has two TV cameras that provide 3D binocular vision to the remote operator. The robot arms and end-effectors are designed to emulate operation of the normal human arms and hands. Tactile feedback will be provided so that the remote human operator will feel pressure, vibration, texture, and temperature. This means that the users will be more comfortable, require less training, and be less likely to make mistakes. This is particularly important since if someone makes a mistake, it may be very difficult to correct since the lab is in space.

Modules may be designed to work independently or in cooperation. As an example of cooperative work would be a series of module designed to produce high quality crystals or integrated circuit chips. In the zero-g and ultra low contamination environment of the RoboLab, it would be routine to produce chips with zero defects. This is particularly important as demand for larger size computer memory grows, especially for chips of gigabyte capacities which are currently not available on earth.

One module may be a stockroom that supplies selected chemicals to a chemical lab module where the chemicals are mixed in correct proportions. This module transfers the mixtures or single elements to a crystal growth module with a furnace. After the crystal is grown it is transferred to a processing module for additional doping. The finished crystal is then transferred to a module which slices and dices the wafer. Next a module packages each die into a chip for testing. Finally a module acts as a storeroom for the completed chips until pickup by astronauts.

VIRTUAL REALITY SIMULATION

The RoboLab virtual reality simulation was developed using special purpose hardware and software. The system I/O components include a Spatial Tracking System and the Data Acquisition and Transmission Unit. The Data Acquisition and

Transmission Unit.includes the VPL EyePhone Model 2 head mounted display and the DataGlove Model 2 hand input device.

The DataGlove is an input device that converts hand motions and flexation into computer readable form. The EyePhone is a stereo color computer display system. Left and right liquid crystal screens show each eye a video image from a slightly different point of view so that the user sees objects in three dimensions.

The EyePhone's headphones provide audio feedback from the virtual reality and the optional AudioSphere System provides three-dimensional real-time sound rendering. The Convolvotron spatializes sounds generated by a MIDI synthesizer.

The image rendering components consists of two Silicon Graphics PowerSeries workstations which run an in-house developed (NASA/JSC Software Technology Branch Lab) real-time rendering package. This is a C program to read data from the Spatial Tracking System, the DataGlove, and simulate the virtual environment in real time by rendering the image and displaying it in the EyePhone head-mounted display.

The software consists of the Solid Surface Modeler for solid-shaded and wireframe 3D geometric modeling. It is used to develop the objects that comprise the virtual environment. The Tree Display Manager is a graphics visualization tool which uses a hierarchical representation of the 3D models created with the Solid Surface Modeler to give structure to the virtual environment.

At runtime, the data acquisition components collect real world information about the user's position and actions. For instance, the DataGlove measures movements in the finger joints while the Spatial Tracking System monitors the head and hands positions and orientation in the real world 3D space.

LONG DURATION LIFE SCIENCE STUDY

One question that has been investigated since the beginning of the space program is — What are the long-term effects of space on living organisms? This question is particularly important as we plan for long duration space flights such as the Mars mission, in-orbit missions such as the space station, and a lunar settlement.

The RoboLab Life Sciences (LS) module is designed to provide some answers to this question. LS is a complete closed ecosystem having plants and animals. The plants are grown using hydroponics gardening and are the food source of the animals. *In-vivo* testing of the animals is performed by the robot which also functions as the gardener of the plants. Through telepresence, the robot plants seeds, fertilizes, and harvests the plants. The produce is fed to the animals. Through blood tests, cell cultures, and a variety of other tests, the health of the animals is determined. The animals will be allowed to breed and most of their progeny will be returned to earth for further testing and studies. However, offspring from each generation will also be kept in LS to observe the long-term effects of space on successive generations.

The plants are chosen for their ease of growth and harvesting. Also, they will provide a valuable source of fresh produce for astronauts in long duration space shuttle flights or on the space station. The initial plants will be lettuce, tomatoes, cucumbers, radishes, peppers, and a variety of herbs. This will allow astronauts to enjoy fresh salads in space. The animals will include rabbits, hamsters, and gerbils. The space-born animals and plant seeds will be returned to earth for testing and then given away by lottery to schoolchildren to raise.

This program imitates the immensely successful tomato seed program in which schoolchildren across the country learned science in an exciting way by raising tomato plants from seeds left up in space for years. The "Astrobunny" program will be even more popular among schoolchildren since these are living creatures. Fig. 3 shows a black and white image of a small RoboLab complex of several lab modules. The actual simulation is in color. The VR hand allows the user to "fly" around in the environment.

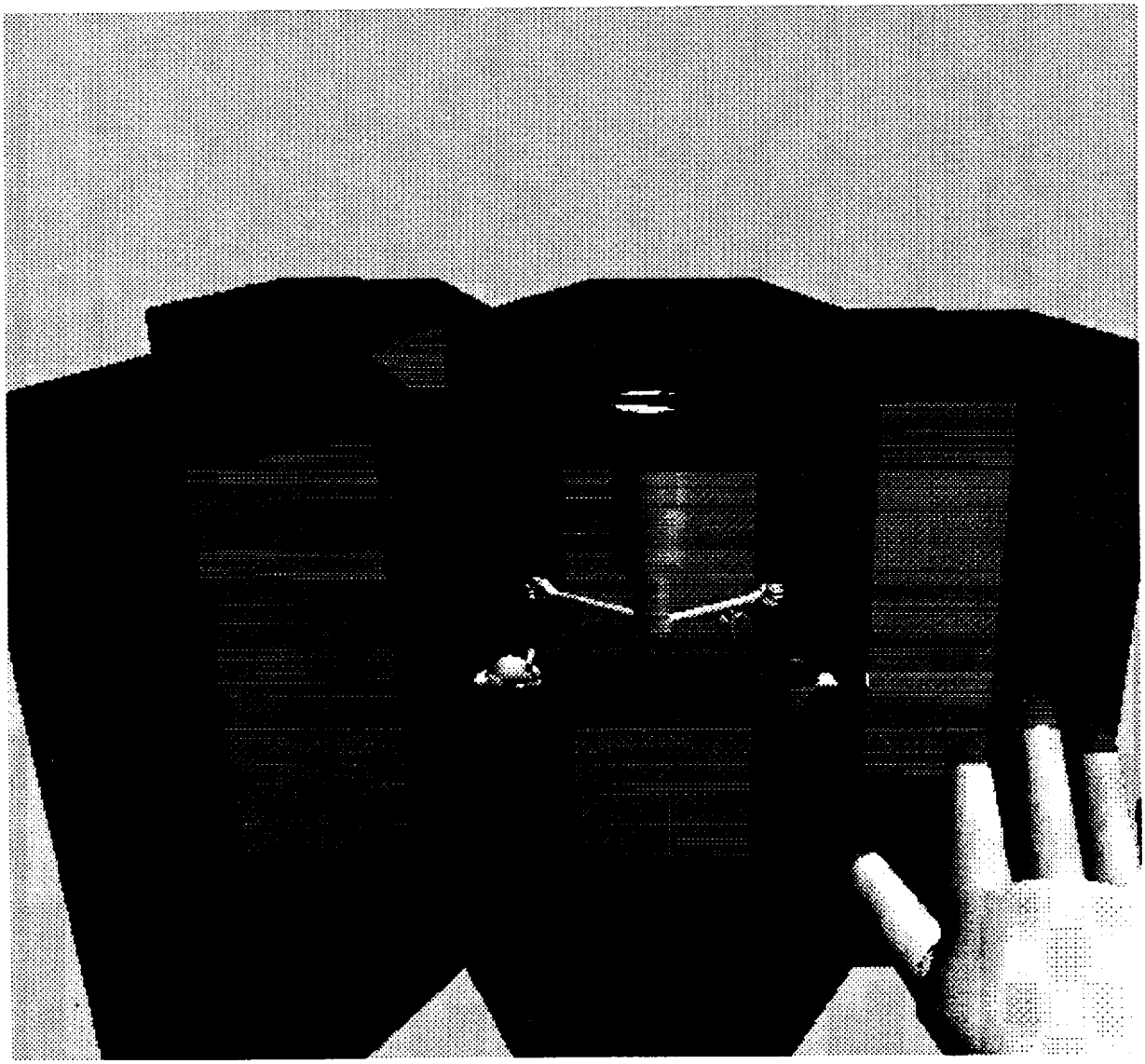


Fig. 3 External View of Several RoboLab Modules in Virtual Reality

Fig. 4 shows an internal view of the Long Duration Life Sciences module. Simulated plants include carrots, lettuce, and tomatoes. Astrobunny is also simulated.

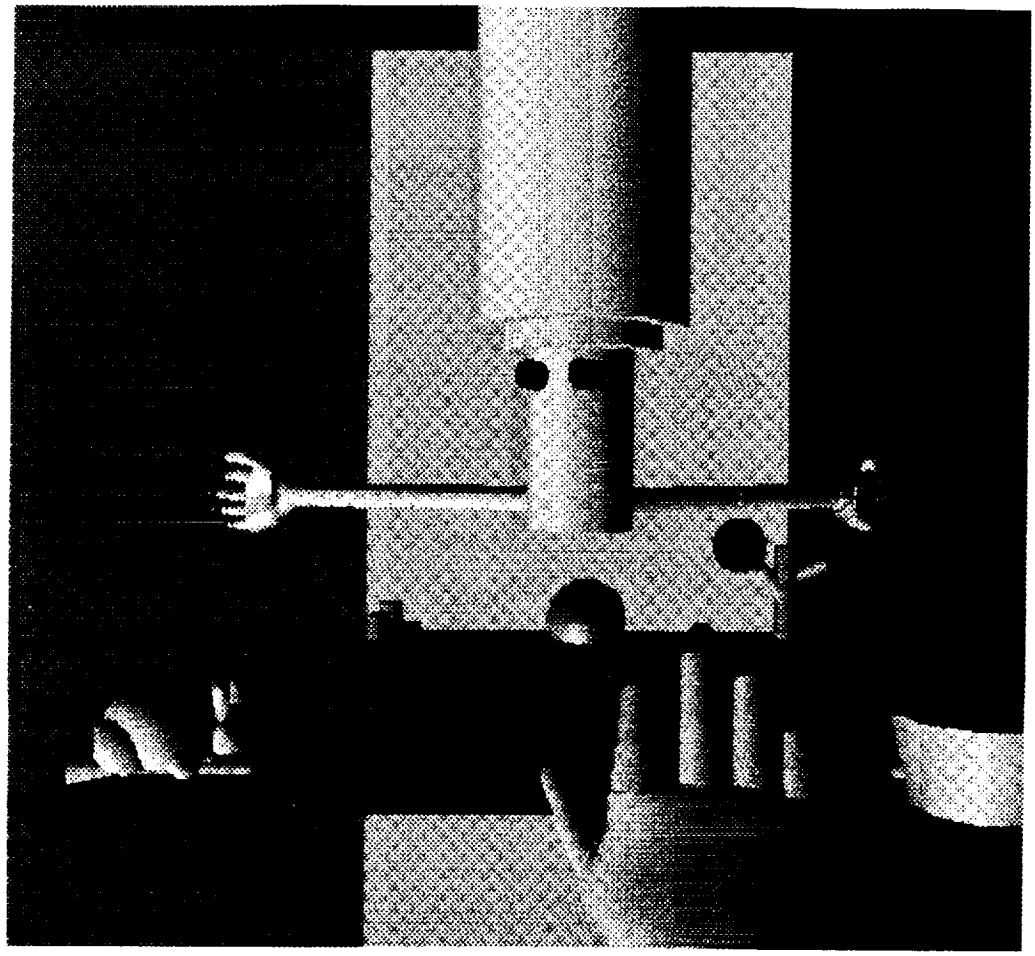


Fig. 4 Internal View of the Long Duration Life Sciences RoboLab

Future plans involve enhancing the RoboLab concept, and adding more modules. As more VR hardware becomes available, we will be able to simulate cooperative RoboLab modules working on joint projects such as semiconductor crystal growing and fabrication. We also plan to develop a RoboLab VR toolkit to facilitate simulation.

Session R5: REMOTE INTERACTION WITH PHYSICAL SYSTEMS

Session Chair: Mr. Joe Herndon

DEVELOPMENT AND DEMONSTRATION OF A TELEROBOTIC EXCAVATION SYSTEM*

B. L. Burks, D. H. Thompson, S. M. Killough, and M. A. Dinkins
Oak Ridge National Laboratory
Robotics & Process Systems Division
P.O. Box 2008, Bldg. 7601
Oak Ridge, Tennessee 37831-6304
Telephone 615-576-7350
Facsimile 615-576-2081

"The submitted manuscript has been authored by a contractor of the U. S. Government under contract DE-AC05-84OR21400. Accordingly, the U.S. Government retains a paid-up, non-exclusive, irrevocable, worldwide license to publish or reproduce the published form of this contribution, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, or allow others to do so, for U.S. Government purposes.

To be presented at the
Space Operations Application and Research Symposium
NASA Johnson Space Center
Houston, Texas
August 3–5, 1993

*Research sponsored by the Office of Technology Development, U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc., and by the Army's Project Manager–Ammunition Logistics under interagency Agreement 1892-AO78-A1 between the Department of Energy and the Armament Research Development Engineering Center at the Picatinny Arsenal.

N94-34028

DEVELOPMENT AND DEMONSTRATION OF A TELEROBOTIC EXCAVATION SYSTEM*

Barry L. Burks
Robotics & Process Systems Division
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 7601
Oak Ridge, Tennessee 37831-6304
Telephone 615-576-7350
Facsimile 615-576-2081

Stephen M. Killough
Robotics & Process Systems Division
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 7601
Oak Ridge, Tennessee 37831-6304
Telephone 615-574-4537
Facsimile 615-576-2081

David H. Thompson
Robotics & Process Systems Division
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 7601
Oak Ridge, Tennessee 37831-6304
Telephone 615-574-5633
Facsimile 615-576-2081

Marion A. Dinkins
Robotics & Process Systems Division
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 7601
Oak Ridge, Tennessee 37831-6304
Telephone 615-574-5710
Facsimile 615-576-2081

ABSTRACT Oak Ridge National Laboratory is developing remote excavation technologies for the Department of Energy's Office (DOE) of Technology Development, Robotics Technology Development Program, and also for the Department of Defense (DOD) Project Manager for Ammunition Logisitcs. This work is being done to meet the need for remote excavation and removal of radioactive and contaminated buried waste at several DOE sites and unexploded ordnance at DOD sites. System requirements are based on the need to uncover and remove waste from burial sites in a way that does not cause unnecessary personnel exposure or additional environmental contamination. Goals for the current project are to demonstrate dexterous control of a backhoe with force feedback and to implement robotic operations that will improve productivity. The Telerobotic Small Emplacement Excavator is a prototype system that incorporates the needed robotic and telerobotic capabilities on a commercially available platform. The ability to add remote dexterous teleoperation and robotic operating modes is intended to be adaptable to other commercially available excavator systems.

*Research sponsored by the Office of Technology Development, U.S. Department of Energy under contract DE-AC05-84OR21400 and by the Army's Project Manager–Ammunition Logistics under interagency Agreement 1892-AO78-A1 between the Department of Energy and the Armament Research Development Engineering Center at the Picatinny Arsenal.

INTRODUCTION

For nearly five decades, the U.S. Department of Energy (DOE) and its predecessor agencies have performed broad-based research and development activities as well as nuclear weapons component production. As a by-product of these activities, large quantities of waste materials have been generated. One of the most common approaches formerly used for solid waste storage was to bury waste containers in pits and trenches. With the current emphasis on environmental restoration, DOE now plans either to retrieve much of the legacy of buried waste or to stabilize the waste in place by in situ vitrification or by other means. Because of the variety of materials that have been buried over the years, the hazards are significant if retrieval is performed by using conventional manned operations. The potential hazards, in addition to radiation exposure, include pyrophorics, toxic chemicals, and explosives. Although manifests exist for much of the buried waste, these records are often incomplete when compared to today's record-keeping requirements. Because of the potential hazards and uncertainty about waste contents and container integrity, excavating these wastes by using remotely operated equipment is highly desirable. In this paper, the authors describe the development of a teleoperated military tractor called the Small Emplacement Excavator (SEE).

The development of SEE is being funded jointly by DOE and the U.S. Army. The DOE sponsor is the Office of Technology Development (OTD), Robotics Technology Development Program (RTDP). The U.S. Army sponsor is the Project Manager for Ammunition Logistics, Picatinny Arsenal. The primary interest of DOE is whose application to remote excavation of buried waste, and while the primary emphasis for the U.S. Army is the

remote retrieval of unexploded ordnance, technical requirements for these two tasks are similar and, therefore, justify a joint development project. Descriptions of this project at an earlier stage have been previously presented (B. L. Burks et al., February 1992, August 1992, and April 1993).

SYSTEM DESCRIPTION

The SEE was chosen as the development vehicle for this project because it is a commercially available system that is already supported by the U.S. Army. Hundreds of SEE units are already in service throughout the world. The goal of the project is to demonstrate the feasibility of retrofitting commercial equipment to achieve high-performance remote operations. SEE is not necessarily the excavator of choice for large-scale waste retrieval campaigns. However, the controls technology developed for SEE shall be readily adaptable to other mechanical systems.

The U.S. Army and DOE perspectives on SEE are different in that SEE modifications may eventually become a moderate-volume production item for the Army, whereas DOE's interest is in more general technology development that will be applied to remote excavation. Hence, within RTDP, development of SEE is part of a larger effort to develop and demonstrate a Remote Excavation System (RES). Because the excavator kinematics, hydraulic control technology, and electronic systems (computers, video, and communications) are similar to backhoes up to large-scale excavators, essentially all the developed technology will be transferable from the telerobotic SEE to the RES program. Although SEE is the specific vehicle that will be used for initial demonstrations of RES controls technology, additional demonstrations are planned to determine and illustrate the degree to which RES controls technology can be readily applied to other excavation platforms.

The SEE vehicle was developed by Freightliner for the U.S. Army for multipurpose use including unexploded ordnance retrieval. SEE has a backhoe on the back and a front-end loader on the front (Fig. 1). The backhoe is an adaptation of the Case 580E commercial backhoe, and the vehicle is a modified Mercedes Benz Unimog truck. Alterations to the vehicle made by Oak Ridge National Laboratory centered upon modifying the hydraulic systems for computer control. High-performance proportional valve components were used to greatly improve the dexterity over the existing manual valves. Proportional valves were chosen rather than servovalves because the former are less sensitive to contaminated hydraulic fluid; also, high-performance proportional valves are now available. Hydraulic pressure sensors provide limited indications of force exerted by the backhoe. Using the pressure data, torque at each joint was computed. The backhoe and front-end loader have also been outfitted with position encoders for use in robotic operations. Remote viewing is provided by two color television cameras with pan-and-tilt mechanisms mounted on the truck body and a third camera mounted on the backhoe boom.

Two productivity enhancement technologies have been deployed on the SEE. As mentioned previously, force feedback was used to give the operator quick feedback of the forces at the shovel. This quick feedback allows the operator to detect many buried objects with which the backhoe comes in contact before the object is uncovered, with the exception of very small or light objects. The second technology was resolved rate control, which allowed the operator to control the motion of the bucket rather than to constantly trade off boom-and-dipper motion to get the desired bucket motion. Industrial excavator vendors are proposing this control system, but none have been implemented on an excavator.

The control station diagrammed in Fig. 2, has been packaged as a portable field unit incorporating two flat-panel video displays and a UNIX-based graphical user interface in two suitcase-sited units. The vehicle's drive system has been modified for remote driving. Only manual transmissions are available for SEE, and because the development of a new transmission is not practical, pnuematic actuators have been installed on the clutch and shift levers to operate the vehicle. Remote steering has been implemented by attaching a hydraulic motor to the steering wheel.

The computer system is an adaptation of an industrial design that is being commonly used within DOE for the RTDP projects. The basic system is composed of a Sun workstation host networked to a VME-based Motorola 68040 target computer, which runs the VxWorks operating system. VME-based computer systems are powerful and flexible because of the wide variety of industrial input/output and powerful single-board computers available.

The communications system between the vehicle and the base station consists of two microwave video channels and an Ethernet data radio. The data radio is a sophisticated spread-spectrum Ethernet packet radio made by Telesystems. Transparent operation of the Ethernet radio enables flexible operation for the computer system. For U.S. Army applications, where a secure communication channel may be required, the option of a fiber-optic bundle has been developed. During the development phase, all computer programs can be downloaded by the radio, thus requiring no software storage on the vehicle. Software management can then be performed solely on the workstation embedded in the console. Near the end of the project, all

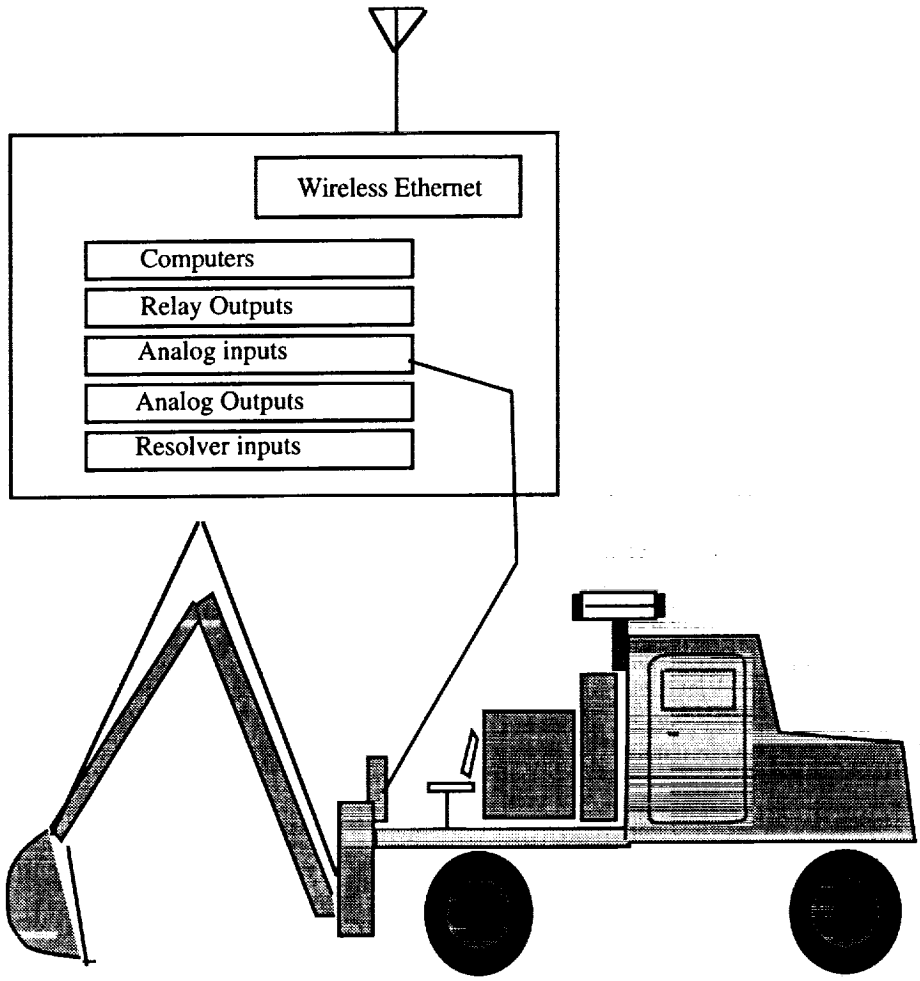


Fig. 1. SEE vehicle and computer interface.

of the software may be put in the computer's read-only memory. The high data rate (one megabaud) also permits teleoperation through the radio link.

Software development is being coordinated with other participants in the RTDP to enable synergistic operation of the various machines for restoration projects. Such coordination activities will involve sharing data between characterization and excavation operations, sharing computer and console resources to reduce expenses, and improving the transferability of collected data and control system code.

Significant improvements to the human-machine interface are featured in the base station to incorporate the data available from characterization activities and present available data from sensors on the vehicle. Computer graphic interfaces are be used to display collected data and aid in vehicle control by presenting vehicle status and position. This human-machine interface has been designed in collaboration with the other remotely driven vehicles in the RTDP to help produce a standardized interface that can be used for several vehicles.

RESULTS

The system was initially demonstrated in December 1992. This first phase involved only remote operation of the backhoe; the vehicle was still manually driven to the work site. The main demonstration focus was feasibility of remotely uncovering waste barrels and digging up contaminated soil or, alternatively, excavating unexploded ordnance.

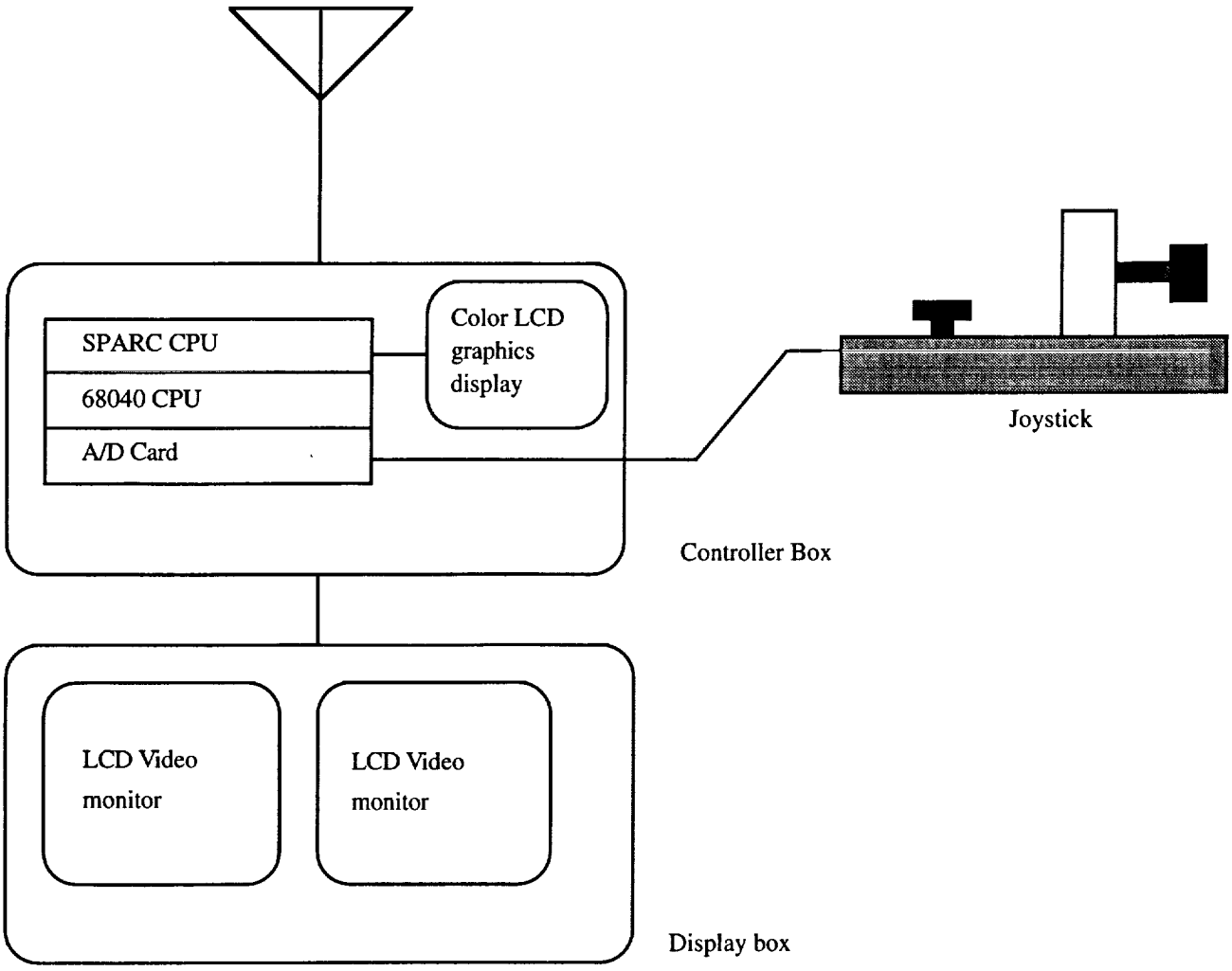


Fig. 2. Hardware architecture of the portable RES Controller.

The second phase of development was completed in the summer of 1993, and involved remote-driving and front-end-loader operations. Demonstrations were performed at the Idaho National Engineering Laboratory, Idaho Falls, Idaho, as part of the OTD Buried Waste Integrated Demostrations (BWID). Some of the results from these BWID demonstrations include comparisons between manual and remote operations for retrieval of a variety of waste container sizes and storage configurations. Demonstrations data are still under analysis, at this writing. However, initial results indicate the SEE, under telerobotic control, provides retrieval capabilities about 1.5 times faster than the same backhoe under manual control for similar excavation scenarios. This is remarkable since teleroperated systems typically require an order-of-magnitude longer for most manipulation tasks than manual operations.

The demonstrations performed from December to July have been extremely valuable in gaining experience in remote excavation, especially the BWID tests. During overburden removal tests a mean depth of within one in. of the desired depth was obtained for shallow digs. The dig depth standard deviation over the 10 ft wide test cell was ± 4 in. The graphical user interface was highly useful for maintaining the position of each backhoe link and location of objects such as the dig and dump areas. With typical teleoperation tasks, a time penalty of a factor of 10 is common. Using the SEE under teleoperation vs manual control a time increase of about 50% was observed for a variety of excavation and waste retrieval tasks. With training, this factor could be further reduced. The intuitive hand controller made operation of the SEE relatively simple, compared to manual operations. A group of novice operators were tested and were found to complete dexterity tasks with 65% accuracy during their first attempt using the SEE.

The third camera on the boom was found to be very useful for "in-hole" operations, in particular. The communications systems were successfully operated with up to one half mile separation between the vehicle and base station.

Additional human factors performance testing will be performed in the fall of 1993 at Redstone Arsenal, Huntsville, Alabama. These studies will allow the same soldiers who routinely operate SEE in manual operations to perform similar excavations by using the teleoperated and telerobotic modes. Field deployment of the telerobotic SEE for military applications will depend greatly on the results of these performance tests.

FUTURE PLANS

Several experimental features are planned for the SEE that will be of potential benefit on remote excavators. The four main experimental areas are robotic operation, new end effectors for the backhoe boom, improved graphics displays, and advanced radio communications.

Several opportunities exist to provide robotic operations that can significantly improve the overall performance of the excavation operation. One envisioned operator improvement is an automatic empty-bucket procedure that will empty the backhoes' load at a preset location. This feature will eliminate the need for the operator to reposition the television cameras for each dumping operation. This feature was implemented for the BWID tests but needs improvement. Another desired feature is robotic gradual excavation of a specified area. This feature would provide both excavation to a precise depth and higher throughput. An additional benefit of robotic excavation would be automatic digging in areas identified as contaminated by other robotic sensors. With such a direct method, the operator would not need to interpret the sensor-data map while operating the backhoe.

Adding to the backhoe the capability of lifting objects as well as uncovering them would be desirable. Ideally, the waste drums could then be lifted out without their contents leaking; thus, the drums could be sealed in a larger new container. Trying to push the drum out with the backhoe scoop would almost certainly damage the drums and spill their contents; therefore, a robotic grappling end effector will be required. Although a separate machine can be used for this task, the preferred option is to provide changeable end effectors for the backhoe. Several end effectors are being studied for this task, the main selection criteria being remote changing of the end effectors and dexterous handling of the drums.

Graphical aids can be used to describe to the operator the current circumstances with respect to vehicle position, area contamination, and excavated areas. Maps of contaminated areas can show the operator where digging operations need to take place. Three-dimensional plots can be used to describe the amount of soil that has been uncovered already and to show the current digging depth. Additionally, three-dimensional graphics can greatly benefit programming and controlling of robotic operations.

Alternate radio communication methods are being investigated because of problems associated with some previous communications schemes. Current microwave video systems perform well but are susceptible to multipath distortion and are poor in over-the-hill performance. They are also quite expensive. Because digital data radios perform much better at lower cost, we are investigating the possibility of digitizing and compressing video so that it may be delivered over a digital link. Technology is advancing rapidly in this area, and we anticipate that digital video transmission will soon become practical at a lower cost.

REFERENCES

1. Burks, B. L., Hannah, J. H., Killough, S. M., and Thompson, D. H., "Development of a Teleoperated Excavator," 19th Annual WATTec Technical Conference and Exhibition, Knoxville, Tennessee (1992).

2. Burks, B. L., Killough, S. M., and Thompson, D. H. "Development of a Teleoperated Backhoe for Buried Waste Excavation," pp. 93–97, Spectrum '92 International Topical Meeting on Nuclear and Hazardous Waste Management, Boise, Idaho, (1992).

3. Thompson, D. H., Burks, B. L., and Killough, S. M., "Remote Excavation Using the Telerobotic Small Emplacement Excavator," pp. 465 , Proceedings of American Nuclear Society Fifth Topical Meeting on Robotics & Remote Systems, Knoxville, Tennessee, April 26–29, 1993.

N94-34029

A TELEOPERATED SYSTEM FOR REMOTE SITE CHARACTERIZATION

Gerald A. Sandness, Ph.D.
Pacific Northwest Laboratory*
Automation and Measurement Sciences Department
POB 999, MS 25
Richland, Washington 99352

Bradley S. Richardson
Oak Ridge National Laboratory+
Robotics & Process Systems Division
POB 2008
Oak Ridge, Tennessee 37831 6304

Jon Pence
Lawrence Livermore National Laboratory++
POB 808, L-363
Livermore, CA 94551

ABSTRACT

The detection and characterization of buried objects and materials is an important step in the restoration of burial sites containing chemical and radioactive waste materials at Department of Energy (DOE) and Department of Defense (DOD) facilities. By performing these tasks with remotely controlled sensors, it is possible to obtain improved data quality and consistency as well as enhanced safety for on-site workers. Therefore, the DOE Office of Technology Development and the US Army Environmental Center have jointly supported the development of the Remote Characterization System (RCS). One of the main components of the RCS is a small remotely driven survey vehicle that can transport various combinations of geophysical and radiological sensors. Currently implemented sensors include ground-penetrating radar, magnetometers, an electromagnetic induction sensor, and a sodium iodide radiation detector. The survey vehicle was constructed predominantly of non-metallic materials to minimize its effect on the operation of its geophysical sensors. The system operator controls the vehicle from a remote, truck-mounted, base station. Video images are transmitted to the base station by a radio link to give the operator necessary visual information. Vehicle control commands, tracking information, and sensor data are transmitted between the survey vehicle and the base station by means of a radio ethernet link. Precise vehicle tracking coordinates are provided by a differential Global Positioning System (GPS).

---

*Operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-AC06-76RLO 1830.

+Operated for the U.S. Department of Energy by Martin Marietta Energy Systems, Inc. under Contract DE-AC05-850R21400.

++Operated for the U.S. Department of Energy by the University of California under Contract W-7405-Eng-48.

The sensors are environmentally protected, internally cooled, and interchangeable based on mission requirements. To date, the RCS has been successfully tested at the Oak Ridge National Laboratory and the Idaho National Engineering Laboratory.

INTRODUCTION

The detection and characterization of waste burial sites require surveys that involve non-intrusive geophysical, radiological, and chemical sensors. Such surveys, performed with manually operated sensors or vehicle-mounted sensors can often detect and map buried objects, materials, contaminants, and geological features to depths of several meters in the earth. Vehicle-based surveys are more efficient than those which involve manual methods, but they have generally suffered from poor vehicle maneuverability and from degraded sensor performance due to interactions with the vehicle. The benefits of vehicle-based sensing can be most fully realized if the survey vehicle is specifically designed to be a sensor platform. Further, a remotely controlled survey system can enhance efficiency and provide a means of safely dealing with sites where it may be undesirable to perform site characterization surveys in which human operators must traverse the site either on foot or on board a survey vehicle.

In Fiscal Year 1992, the U.S. Department of Energy's Office of Technology Development (OTD) initiated the development of the Remote Characterization System (RCS). The primary objective of this continuing project is to develop a remotely controlled system that can perform site characterization surveys that will be safer and more cost effective than those that are being performed by other available methods. At the same time, it is expected that the data sets produced by the RCS should be at least as accurate and complete as those produced by other survey systems. The remote-control capabilities of the RCS will improve safety at hazardous sites by reducing on-site manpower requirements and by minimizing the exposure of personnel to unnecessary risks. It is also expected that RCS subsystems will be utilized in other DOE telerobotic applications to achieve time and cost savings in other phases of site cleanup. The vehicle tracking capability of the RCS has already been transferred to a teleoperated excavation system that has been developed at the Oak Ridge National Laboratory.

The major hardware and software components of the prototype system have now been developed and assembled. Initial system tests have been performed at test sites at the Oak Ridge National Laboratory and at the Idaho National Engineering Laboratory. Additional tests at waste burial sites and technology transfer of the RCS are planned for FY 1994.

Joint support for this work has been provided by the U.S. Army Environmental Center. The project is a collaborative effort involving the Pacific Northwest Laboratory, the Oak Ridge National Laboratory, the Sandia National Laboratory, the Lawrence Livermore National Laboratory, and the Idaho National Engineering Laboratory.

SYSTEM OVERVIEW

The RCS design philosophy required that the remotely controlled survey vehicle and its instrumentation be small, light, and relatively inexpensive. Another requirement was that the vehicle must be constructed predominantly of non-metallic materials so that it will have a minimal effect on the operation of on-board geophysical sensors. The suite of sensors supported by the vehicle and its instrument package currently includes ground-penetrating radar (GPR), a metal detector, a magnetometer, a magnetic gradiometer, an induction-type ground conductivity sensor, and a radiological sensor.

Figure 1 is a drawing of the system in a field application. Although the picture differs from the actual system in certain details, it illustrates the basic system configuration. The vehicle is self-propelled and is guided by an operator located at a remote base station. Telemetered video signals give the operator the visual information needed to control the vehicle. Digital commands for vehicle and instrument control are transmitted to the vehicle. Data produced by the on-board sensors are transmitted from the vehicle to the base station where they are recorded, processed, and displayed.

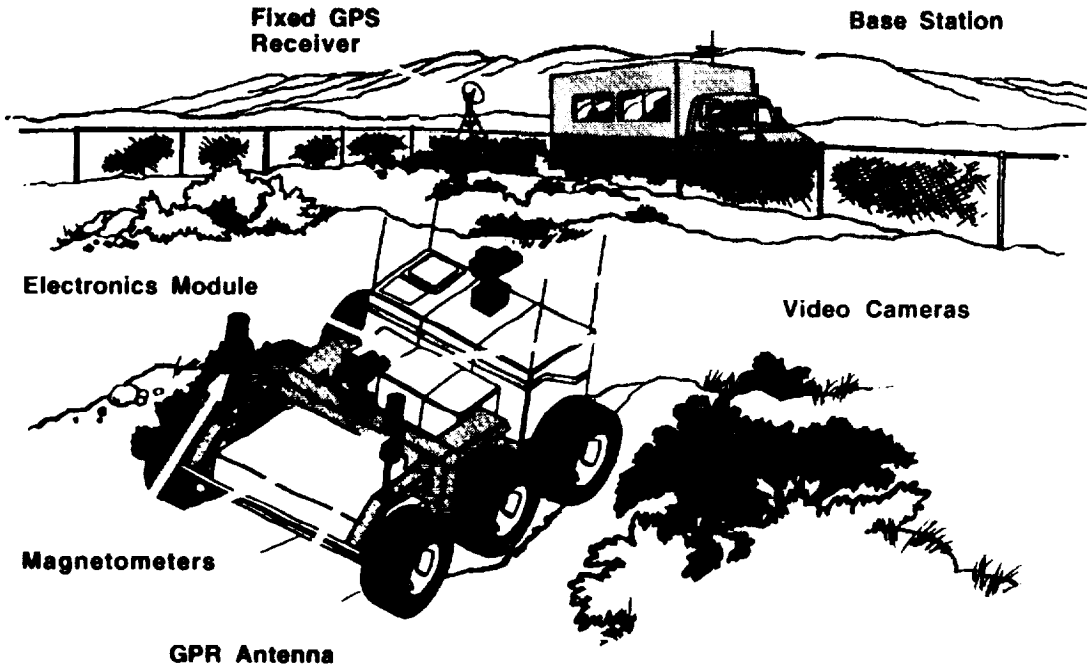


Figure 1. Drawing of the RCS.

THE SURVEY VEHICLE (LSV)

The construction of a sensor-compatible low-signature vehicle (LSV) required the use of a minimum amount of metallic material. The current prototype vehicle contains approximately 130 lbs of metal, but this material is distributed so that it has only a small effect on the on-board geophysical sensors. The most critical part of this effort was to reduce the amount of

magnetic material (steel) on the vehicle and to locate unavoidable steel components as far from the magnetometers as possible.

A typical site for a geophysical field survey exhibits surface features such as bushes, trees, fences, buildings, parked vehicles or other machinery, open holes, depressions, ditches, hills, berms, rocks, and miscellaneous debris (wire, cable, 55-gal drums, concrete blocks, etc). To obtain the maneuverability needed to operate the LSV among these kinds of obstructions, we adopted two additional design requirements. First, the LSV must be able to turn in place. Second, all sensors and other vehicle components must be contained within the perimeter of the vehicle as defined by its wheels and bumpers. These requirements eliminated the possibility of transporting sensors on a trailer or a boom. In particular, the large size of a ground-penetrating radar antenna and the necessity of coupling it to the ground virtually dictated that the vehicle be designed around it. Thus, as illustrated in Figure 1, the front part of the chassis is an open structure that permits the GPR antenna to be suspended between the front wheels.

Figure 2 is a photograph of the prototype LSV that has been constructed at the Pacific Northwest Laboratory. This vehicle is approximately 7 ft long and 5 ft wide. Its weight is approximately 800 lbs, including a payload of approximately 150 lbs. Its major components include the chassis, the engine, the drive train, and an electrical power generator. They also include an on-board digital controller and peripheral devices to monitor vehicle status and to provide low-level control inputs to the vehicle.

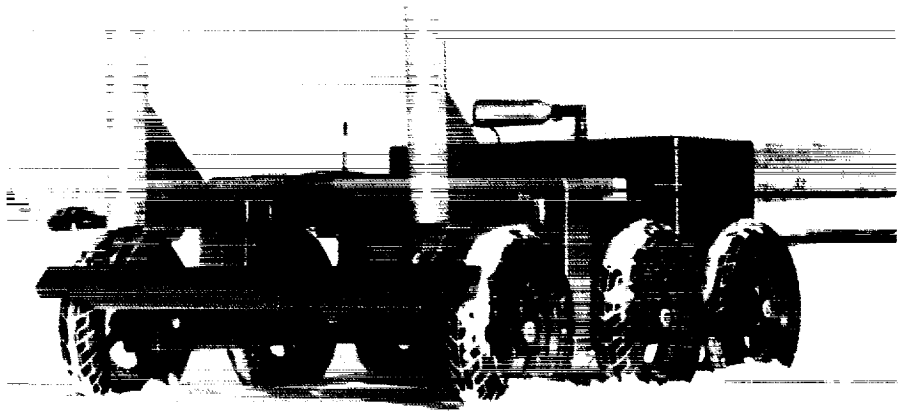


Figure 2. The RCS Low-Signature Vehicle.

The LSV is based on a six-wheeled design with modified skid steering. To equalize wheel loading and to minimize the vertical movement of the instrument platform in response to the roughness of the ground surface, we developed a simple articulated chassis that has proven to be very effective. It consists of two main sections that form the rear third and the forward two-thirds of the vehicle, respectively. A pivot located on the vehicle's longitudinal axis allows the the front and rear sections of the chassis to rotate relative to each other. Additional articulation is provided at the front end of the

chassis. The two wheels on each side of the front section of the vehicle are mounted at the ends of a horizontal arm. Each of the two arms is connected by a bearing to the ends of a yoke, or inverted U-shaped member, that straddles the front part of the chassis. Each arm is free to pivot about a transverse axis located at the center of the arm.

A 20-hp, gasoline-powered, 2-cylinder engine is mounted on the rear section of the chassis. A 12-V, 50-amp alternator mounted on the engine provides electrical power for the sensors, control modules, and other electronic devices on the vehicle. A hydraulic pump, electronically controlled hydraulic valves, and four hydraulic motors provide power at the front and rear wheels.

The LSV has been designed to climb and traverse 35° slopes, to have a ground clearance of 8 in. (except for the GPR antenna), and to operate at speeds up to 5 ft/s. These features permit operations on most of the terrain present at DOE and DOD waste burial sites.

NAVIGATION SUBSYSTEM

A differential kinematic implementation of the satellite-based Global Positioning System (GPS) is the primary means of tracking the LSV. The differential configuration involves the use of two NovAtel (Calgary, Alberta, Canada) GPSCard Model 951R receiver modules. The first, mounted on the LSV, computes its location and transmits that information to a dedicated computer in the RCS base station using an embedded computer and telemetry unit. The second module is mounted on the base-station truck. It is fixed in position for a given survey and provides error-correction information that is transmitted to the LSV's GPS receiver. Coordinates accurate to ±50 cm (typically) are calculated in real time at a rate of 5 measurements/s. Coordinates accurate to ±15 cm (typically) are obtained by post-processing the recorded GPS data.

COMMUNICATIONS SUBSYSTEM

A digital, radio-frequency (RF), command/data link provides ethernet communications between the vehicle and the base station. Signals transmitted to the LSV control the direction and speed of the vehicle, the orientation of the video cameras, and the setup and operation of the on-board sensors. Vehicle status information and sensor output data are transmitted from the LSV to the base station. Setup commands are transmitted to each sensor prior to the initiation of a survey, and parameter update commands can be transmitted to the sensors at any time. After data collection has been initiated, the sensor data are transmitted at predetermined intervals without intervention or commands from the base station. This approach permits data to be transmitted at 25 kbytes/s, a rate sufficient to handle the 17-kbyte/s output of the GPR sensor together with the output of all of the other sensors. Two separate analog RF channels handle video transmissions.

HIGH-LEVEL CONTROL STATION (HLCS)

The operator interface to the LSV is called the High-Level Control Station (HLCS). It is contained in the base-station vehicle and communicates with the LSV via the RF telemetry link described above. The components of the HLCS are housed in the truck shown in Figure 3. The cargo box was custom

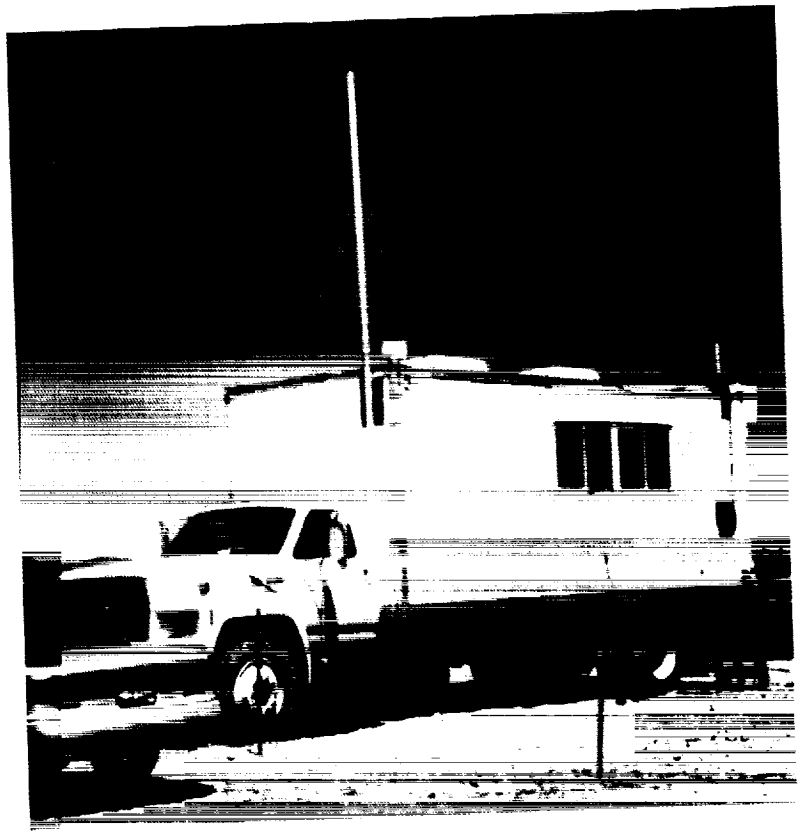


Figure 3. The truck housing the RCS base station.

built to provide equipment mounting space, electrical power, lighting, heating, air conditioning, windows, counter space, and storage cabinets.

The HLCS provides the hardware and software for remote driving (teleoperation), camera positioning, and data displays. A central feature is a control chair with vehicle joystick controls and a keyboard/trackball interface for command inputs to the graphics-based operator interface (Figure 4). The system operator sits in the control chair, driving the remote vehicle and controlling the video cameras with joysticks and fingertip controls. The remote video images and a graphical interface to the control computer are presented on video displays located in front of the operator. The operator also controls sensor selection, sensor operation, and data acquisition through the graphical operator interface. A secondary graphical data display station is provided to allow a geophysicist or observer to examine real-time data. Planned extensions of the control features emphasize automated and semi-automated survey capabilities that will reduce he burden on the operator. An additional potential extension would provide multiple vehicle control by one station with occasional operator input during problem resolution.

Figure 4. The operator's control station.

VIDEO SUBSYSTEM

The system operator must receive visual information from the LSV so that he can recognize hazards and obstructions and can guide the vehicle around them. It is vital that the information available to the operator be sufficiently detailed that he can make on-the-fly decisions regarding the risks associated with anomalous features that the LSV will encounter in the field. A stereo video subsystem is planned to provide the necessary detailed visual information, but the current configuration provides two monoscopic channels that are set up for viewing in the forward and backward directions. The current system includes the cameras, camera control components (pan/tilt), and the associated telemetry links needed for stereo viewing, but does not include the necessary stereo display and head-tracking components. These, together with a data compression technique that will permit both video channels to be transmitted on a single RF link, represent goals for system improvement.

SENSORS

To date, the following sensing instruments have been mounted on the LSV for testing:

- Fluxgate magnetic gradiometers (Model APS-511, Applied Physics Systems, 897 Independence Avenue, Mountain View, CA 94043)
- Cesium vapor magnetometers (Model G822A, EG&G Geometrics, 395 Java Drive, Sunnyvale, CA 94089)
- Sodium iodide gamma detector (2-in. thick, 5-in. diameter crystal, Harshaw/Filtrol, 6801 Cochran Road, Solon, OH 44139).
- Ground-penetrating radar (Model SIR 3, Geophysical Survey Systems, Inc., 13 Klein Drive, North Salem, NH 03073-0097)
- Electromagnetic induction ground conductivity sensor (Modified Model EM31, Geonics Ltd., 1745 Meyerside Drive, Unit 8, Mississauga, Ontario, Canada L5T 1C5)

It has been proposed that a portable mass spectrometer under development at the Lawrence Livermore National Laboratory be added to this package to provide a chemical sensing capability. Not all of the sensors will be mounted on the vehicle at any given time. This is partly due to inherent differences in operating requirements or operating modes. In particular, for radiological and chemical sensing, the vehicle will probably be operated at a low speed or in a slow start-stop mode rather than the fast continuous-motion mode that is appropriate for the geophysical sensors.

The test data sets that have been collected to date, are currently being processed, but initial results are available for the magnetic and radiation sensors. Figure 5 is a contour map that illustrates the data produced by the cesium vapor total-field magnetometer. This data set was recorded at an uncontaminated (cold) test pit at the Idaho National Engineering Laboratory. It compares favorably to equivalent data sets collected by manual methods. The locations of the magnetic anomalies shown in this figure correspond well to known locations of buried objects. Repeated measurements over the same sets of test objects have shown that the data produced by the LSV-mounted magnetic sensors and the GPS tracking subsystem are both stable and repeatable. Figure 6 shows an orthographic projection of radiation intensity data produced by the sodium iodide gamma ray sensor. The radiation source for this test survey was a small packet of lantern mantles buried just below the ground surface.

A project is currently underway at the Pacific Northwest Laboratory to develop a compact, rugged, high-performance, ground-penetrating radar system that can be operated in a remotely controlled mode. However, the sensors currently deployed on the LSV are commercially available instruments. Modifications are being made to minimize their size, weight, and electrical power requirements and to improve their ruggedness. Each sensor includes a small embedded computer that provides interfacing to the RCS communications network.

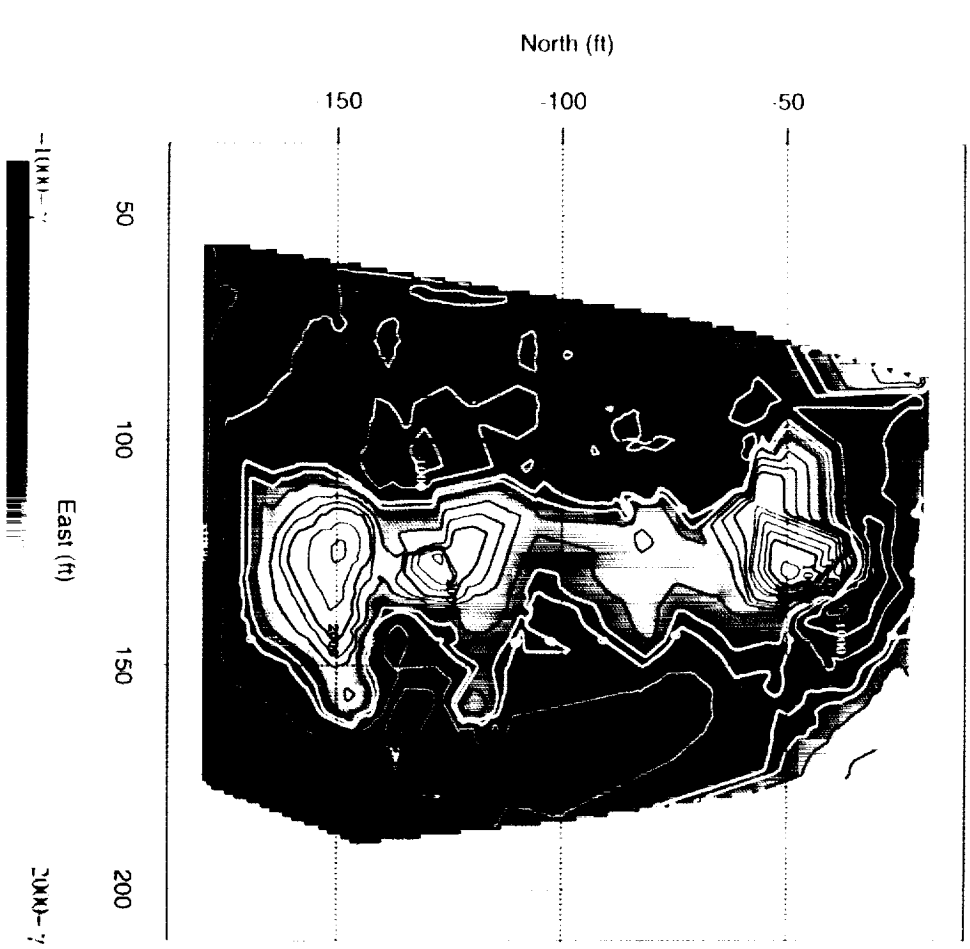


Figure 5. Total-field magnetic contour map.

CONCLUSIONS

Initial tests of the prototype system have shown that the system will provide the desired benefits of enhanced safety, efficiency, and data quality in site characterization operations. The ability of the GPS subsystem to provide accurate vehicle and sensor coordinates is particularly significant because automated tracking is a crucial factor in telerobotic operations at hazardous sites. The display of video, compass heading, and real-time GPS tracking data on the operator's console allows the operator to drive the survey vehicle accurately along desired survey paths. In addition, the real-time display of sensor output on a data display monitor allows the operator to identify features of particular interest and to ensure that the track spacing adequately delineates those features. The efficiency of the survey operation and subsequent data processing procedures is enhanced by the ability of the RCS to acquire multiple data sets simultaneously and to attach time stamps and geographical coordinates to each datum.

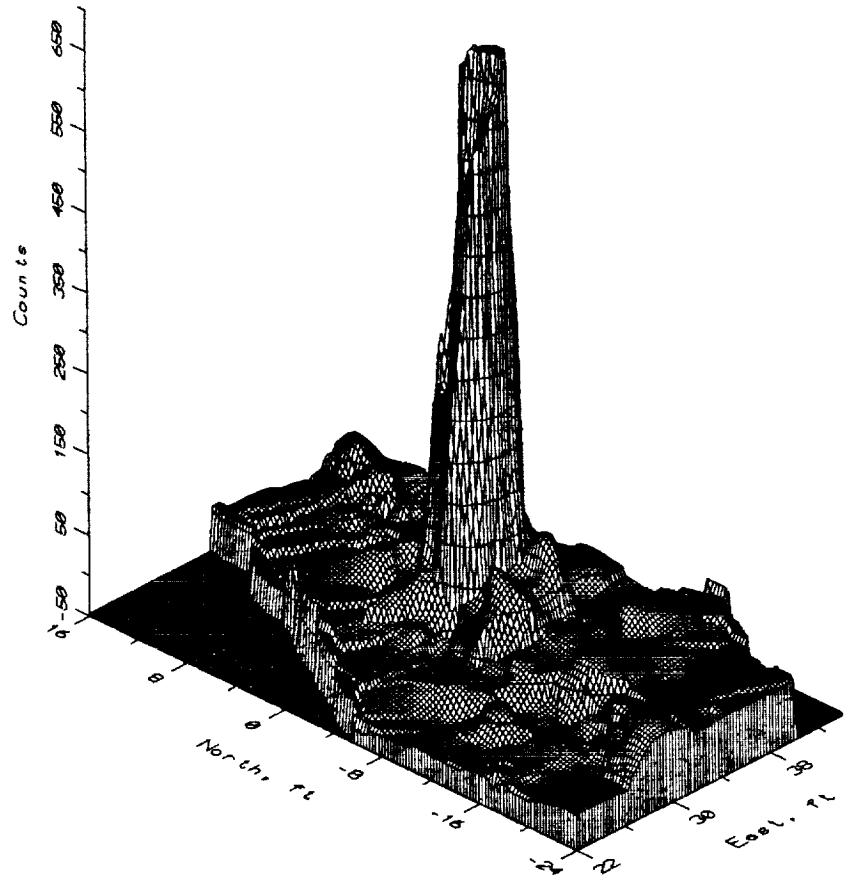


Figure 6. Orthographic projection of gamma radiation intensity from a localized source.

Although the metallic content of the LSV has not yet been reduced to the desired minimum level, the vehicle has proven to be an effective low-signature platform for the magnetic, radiological, and GPR sensors. The principal effect of the LSV's engine and the other metallic drive train components has been a reduction in the stability and effective sensitivity of the EM31 electromagnetic induction sensor. Efforts are currently underway to improve the performance of that sensor. A continuing objective of the RCS project is to further reduce the number of metallic components on the vehicle.

One of the proposed operational functions of the RCS is to work in parallel with waste site excavation equipment in what is called the "scratch and sniff" mode. This mode involves repetitive site characterization surveys as layers of overburden are removed from the waste deposit. As the chemical and/or radiological contaminants are progressively exposed, the RCS will be able to define and characterize the waste materials with increasing levels of detail and accuracy without exposing human operators to the hazards associated with proximity to the waste materials. In this mode, data relating to the distribution of waste materials and contamination levels will be used to formulate and refine excavation strategies.

N94- 34030

Vehicle Development of Lunar/Mars Exploration

James W. Purvis
Sandia National Laboratories
Albuquerque, NM

The author of this paper presents a historical discussion of robotic vehicle development of lunar and martian exploration. The discussion begins by comparing and contrasting the transportation, environmental, and operational requirements on the two planets. This is followed by a historical summary of what has been done to date, including some recently released information on the Soviet rovers sent to Mars and Phobos in the early 1970s. Finally, current proposed missions, vehicles, operational requirements, and development status are discussed.

CONTROLLING TELEROBOTS WITH VIDEO DATA AND COMPENSATING FOR TIME-DELAYED VIDEO USING OMNIVIEW™

Dan Kuban, Steve Zimmermann & Lee Martin
TeleRobotics International, Inc., 7325 Oak Ridge Highway, Knoxville, TN 37931

ABSTRACT

Remote viewing is critical for teleoperations, but the inherent limitations of standard video reduce the operator's effectiveness. These limitations have been compensated for in many ways, from using the operator's adaptability, to augmenting his capability with feedback from a variety of sensors and simulations. Omniview™ can overcome some of these limitations and improve the operator's efficiency without adding additional sensors or computational burden. It can minimize the potential collisions with facility equipment, provide peripheral vision, and display multiple images simultaneously from a single input device. The Omniview™ technology provides electronic pan, tilt, magnify, and rotational orientation within a hemispherical field-of-view without any moving parts. Image sizes, viewing directions, scale and offset etc., may be adjusted to fit operator needs.

This paper discusses the derivation of the image transformation, the design of the electronics, and two applications to telepresence that are under development. These are Video Emulated Tweening (VET), and Manipulator Guidance and Positioning (ManGAP). The VET effort uses Omniview™ to compensate for time-delayed video in teleoperation of remote vehicles. In ManGAP two Omniview™ systems are used to provide two sets of orientation vectors to points in the field-of-view (FOV). These vectors then provide absolute position information to both control the position of a telerobot, and to avoid collisions with the work sight equipment.

INTRODUCTION

Remote viewing is the most critical feedback in teleoperations. Close viewing is necessary for detailed manipulation tasks, while wide-angle viewing aids the positioning of the remote handling system and helps avoid collisions in the work space. The majority of these systems use either a fixed-mounted camera with a limited viewing field, or they utilize mechanical pan-and-tilt platforms and mechanized zoom lenses to orient the camera and magnify its image. These mechanisms can be large, unreliable, and may interfere or collide with the environment. Also, several cameras may be necessary to provide wide-angle viewing or complete coverage of the work space. Camera viewing systems that use prisms or mirrors to provide wide viewing angles have been developed in order to minimize the size and volume of the camera and minimize the amount of intrusion into the viewing environment, but this approach can result in blind spots. Also, these systems typically have no means of magnifying the image and or producing multiple images from a single camera.

The Omniview™ solution is based on the property that a fisheye lens allows a complete hemispherical field-of-view to be captured, but with significant barrel distortion present in the image periphery. A high speed image transformation processor has been developed that reconstitutes portions of the image to correct the lens distortion for display on an RS-170 standard format monitor. The Omniview™ imaging system has several advantages over standard camera systems. Multiple images may be simultaneously produced by the device allowing a single omnidirectional camera to provide numerous independent views from one location. The transformation is accomplished electronically, providing complete programmable control over viewing parameters.

IMAGE TRANSFORMATION

The postulates and equations for transforming the input image are based on the camera system

utilizing a fisheye lens as the optical element. There are two basic properties and two basic postulates that describe the perfect fisheye lens system. The first property of a fisheye lens is that it encompasses a 2π steradian or hemispherical field-of-view and the image that it produces is a circle. The second property of the lens is that all objects in its field-of-view are in focus, i.e. the perfect fisheye lens has an infinite depth-of-field. In addition to these two main properties, the two important postulates of the fisheye lens system are stated as follows:

Postulate 1: Azimuth angle invariability - For object points that lie in a content plane that is perpendicular to the image plane and passes through the image plane origin, all such points are mapped as image points onto the line of intersection between the image plane and the content plane, i.e., along a radial line. The azimuth angle of the image points is therefore invariant to elevation and object distance changes within the content plane.

Postulate 2: Equidistant Projection Rule - The radial distance, r, from the image plane origin along the azimuth angle containing the projection of the object point is linearly proportional to the zenith angle ß, where ß is defined as the angle between a perpendicular line through the image plane origin and the line from the image plane origin to the object point.

Using these properties and postulates, the mathematical transformation for obtaining a corrected perspective image can be determined. These have been reported previously.[1] By knowing the desired zenith, azimuth, and object plane rotation angles and the magnification, the corrections to the input image can be calculated. This relationship provides a means to transform an image from an input image memory buffer to an output image memory buffer exactly. Also, the fisheye image system is completely symmetrical about the zenith; therefore, the vector assignments and resulting signs of various components can be chosen to reflect the desired orientation of the object plane with respect to the image plane. In addition, the transformation can be modified for various lens elements as necessary for other fields-of-view.

SYSTEM DESCRIPTION

The system consists of a wide angle lens, camera, Omniview™ transformer, display controller, and video monitor. The system is designed to be independent of the camera/lens and monitor and can be used with CCD or tube cameras, visible or infrared spectrums.

A block diagram of the prototype system is shown in Figure 1. The camera input image capture electronics uses a parallel RS-485 type interface to capture the output of the camera. The input and output image memory buffers consist of video RAM arrays with 8 bit resolution. The output display electronics provides a gray-scale 60 Hz interlaced display for an RS-170 standard display monitor. The 80C196 core provides the control interface functions for the prototype system as well as the calculation of the coefficients and parameters for the image transformation core. The trigonometric functions (sin,cos,tan) were implemented using a lookup table with resolution to within a degree. This was found to be sufficient since the direction-of-view parameters are input to the camera system as direct angles for pan, tilt, and rotation. There are two independent processor channels that calculate the corrected pixel positions corresponding to the mapped input coordinates for each direction-of-view. The image transformation processor is pipelined using both high speed arithmetic devices and FPGA elements in order to maximize overall performance.

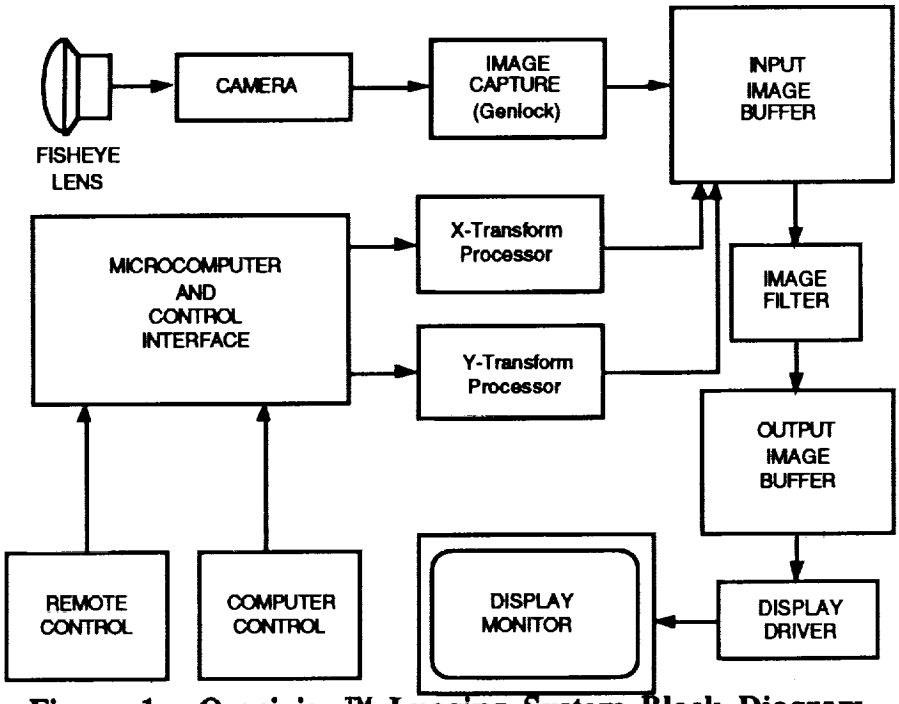


Figure 1 - Omniview™ Imaging System Block Diagram

APPLICATION OF OMNIVIEW™ TO TELEPRESENCE

The Omniview™ technology has many applications in remote viewing and telepresence. Two such development activities are currently underway at TRI. These are Video Emulated Tweening (VET), and Manipulator Guidance and Positioning (ManGAP). The VET effort uses Omniview™ to compensate for time-delayed video in teleoperation of remote vehicles. It relies on Omniview™'s capability to reorient the image without moving the camera to provide the operator with virtual video frames in between the real frames.

VET

A number of space related and teleoperated activities involve the transmission of slow-scan images (image updates slower than the standard 30 frames per second) due to transmission bandwidth or distance. When the slow-scan image is combined with direct operator interaction (for moving a vehicle, for manipulating an object, or for docking two vehicles) the operator often has to employ a "move and wait" strategy to overcome the delays associated with the video update. Of the methods used to counteract this problem, the most common approach presently under development involves predictive graphic simulation of the environment for projecting future actions.

The Omniview™ provides the ability to reorient the camera image without any motion of the camera or its video output, giving the operator the perception that the camera is moving. In practice, the perception of motion can be generated by modifying the pointing angle or magnification values in the transformation. Panning and tilting the image emulates turning and climbing, while magnifying and rotating emulates forward motion and tipping. For example, by matching the vehicle forward speed to the magnification, the operator can perceive vehicle motion by only manipulating the video image. This virtual motion has been demonstrated by using an enlarged aerial photo to simulate flight. The VET seeks to unite this perception of motion with the vehicle characteristics to provide an accurate and realistic emulation of continuous vehicle teleoperation with time delayed-video.

VET creates real-time intermediate video frames from live slow scan video based on vehicular motion commands. Utilizing slow scan video input, it captures the most recent image and adjusts the perspective in real time based on drive commands to the vehicle. The prototype system provides the operator with 22 frames/sec video yielding the perception of non-delayed communications through virtual video during the "delay interval".

This objective has been demonstrated on a vehicular viewing/operation system using Omniview™ with a slow scan video input (1 frame every four seconds) and vehicle control inputs to control the pan, and zoom of the image. The video camera is mounted on the remotely controlled vehicle. Figure 2 shows the image updates source verses time for real time video and for simulated video. At each live interval, a new video frame is captured and displayed, but some 100-140 intermediate frames are generated in between these real frames. For the ground vehicle demonstration development (a radio controlled car), two parameters were varied. The forward vehicular motion was simulated by zooming the image, and the turning of the vehicle was simulated by panning the image.

The match between the last simulated frame and the next live frame must be reasonable to insure that the operator does not receive a disturbing discontinuity in the displayed video. Live full frame video from the moving camera was recorded and compared to that produced with the slow-scan video and VET. Comparison of these two video results indicates that the degree of matching performed by the simulation relative to the actual image is sufficient to convince the operator that he has continuous motion. The effort surpasses a graphically generated approach by using the actual video image as the foundation for the tweening simulation of the remote vehicle, without the computational burden associated with graphic manipulation. The results are not only applicable to remote vehicular operation, but also to robotic teleoperation and spacecraft docking maneuvers.

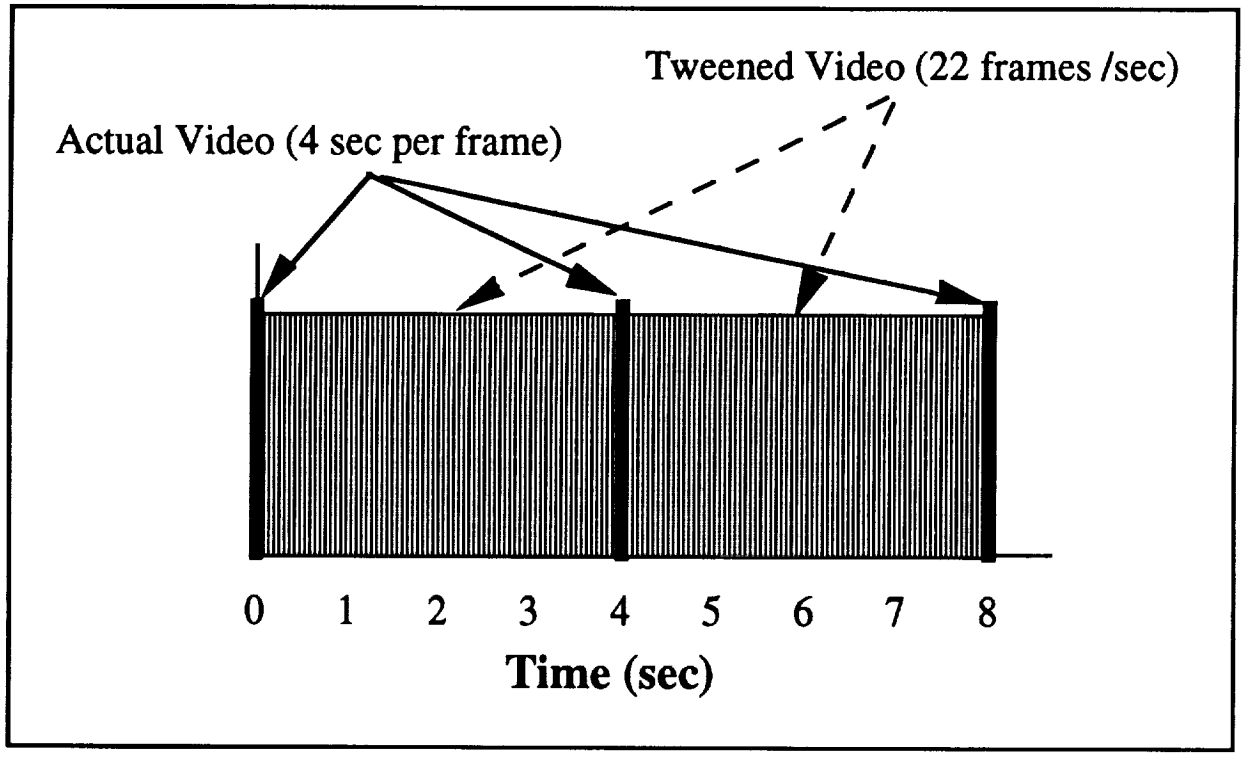


Figure 2 - Image update versus source for transmitted image and emulated image.

System block diagram is in Figure 3. The user interface and simulation subsystems obtain commands from the operator to control the vehicle, and then use those same commands to model the vehicular motions. Two information paths are initiated by the vehicle radio control transmitter. One path controls the vehicle via an RC link. The second RC link path actuates servos that allow the commands that are being sent to the vehicle to be monitored and read by the simulation subsystem. This second RC path receives the control signal, drives servos similar to the ones on the vehicle, converts the mechanical movement of the servo to an electric voltage via a linear potentiometer, and then samples this voltage using an analog to digital converter.

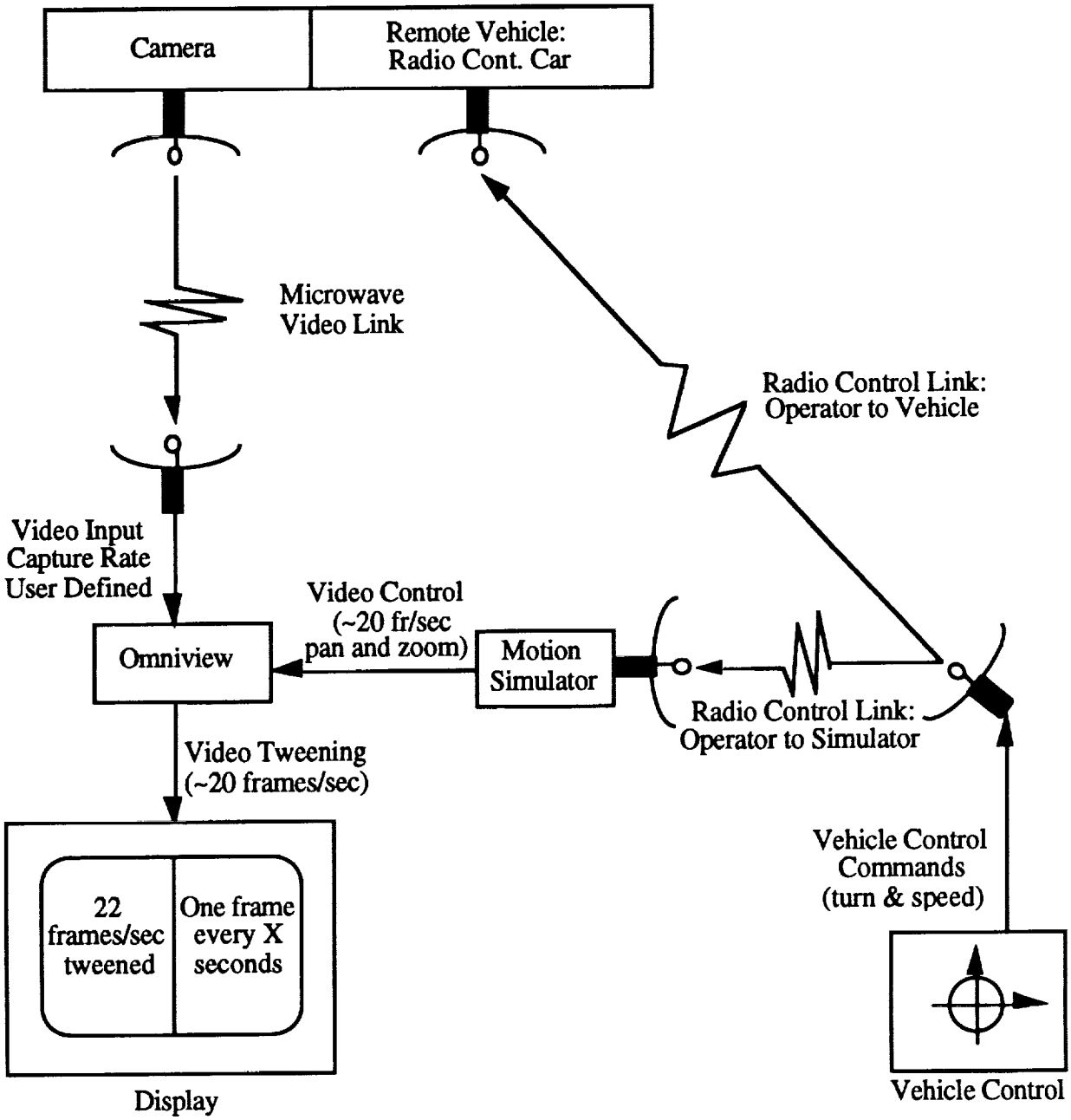


Figure 3 - Block Diagram of Phase 1 Implementation.

The user interface has been configured so that the key simulation parameters can be modified from a series of input switches. Using these switch inputs with observations of the vehicle and the images from it, an opportunity to empirically tune the video tweening model is provided during vehicle operation. For example, the effect of the zoom gain is to make forward motion appear to be occurring even though the input video is frozen due to slow scan time delays. As the zoom gain is increased, the vehicle appears to be moving at a higher rate of speed. At some point, the future simulated image and present actual image will converge. If the zoom gain is too low, the transition from future simulated to present actual images will appear to lurch forward for an instant. Conversely, if the zoom gain is too high, the simulated vehicle appears to move faster than the actual vehicle causing a reverse jump at the transition. If the magnification rate is matched then there is a smooth transition from the last virtual image to the next real frame, achieving the desired VET effect.

It works! In a very qualitative sense, Video Emulated Tweening (VET) achieves virtual reality. It gives the operator the perception of motion even though a still image is all that is available as input to the system. The transition between last virtual frame and first new image was not totally seamless in the prototype implementation, but the possibility for seamless performance exists if a reasonable knowledge of the relationship between the video source and the principle objects in the field are known, and if interlacing effects are eliminated through further development of Omniview™.

ManGAP

The second telepresence development activity that takes advantage of Omniview is the Manipulator Guidance and Protection (ManGAP) system. This system will be implemented and tested as part of the Integral Fast Reactor Program at the Argonne National Lab- West (ANL-W). This effort is driven by experience from operating remote facilities that has shown that transporting and positioning of remote handling equipment typically requires in excess of 50% of the total task completion time. The ManGAP applies video data from Omniview™ to minimize operator effort in the positioning of the teleoperator and transporter system.

The Omniview™ transformation is based on the orientation vector of the direction of interest relative to the camera axis. The three orientation angles of pan, tilt, and rotation to any point in the field-of-view are available from the Omniview™ processor. By selecting a point in the field-of-view on the monitor, the three orientation angles to this point relative to the camera axis are known. If a second Omniview™ is used and offset from the first, and the same point in the field-of-view is selected on the second monitor, then a second set of orientation angles is known. With a fixed offset between the two Omniview™ cameras, these two orientation vectors can be used to triangulate the X,Y,Z position of the selected point relative to the cameras. For controlling a teleoperator, the system requires a fixed location of the teleoperator relative to the Omniview™ pair, and the inverse kinematic transformations for the arm.

In the ManGAP system two Omniviews™ will provide plan view and front view coverage. The operator will use the front view to select the destination of the next motion, and utilize the plan view to designate the distance to the ending location. In this way, the operator will be able to fly the teleoperator end-effector or transporter to an end location by simply selecting the destination on two monitors. The ManGAP block diagram is shown in Figure 4.

A second realm of operation is also being developed - video based collision avoidance. In this mode the control system determines the direction for movement, redirects the manipulator along the line of movement and initiates a sequence of motion constraints to minimize the potential of collision between the manipulator and the working environment.

In this mode the operator determines a geometric area on the first monitor by drawing a graphical square, rectangle or circle. In the second monitor, he indicates a depth, thereby determining a volumetric boundary or envelope. With the Omniview™ orientation vector data, the location and size of this envelope are determined. The operator can define this envelope as a "keep-out" zone, or a "safe" zone. The ManGAP control system will then constrain the teleoperator to "stay out of" or "stay inside" this geometric envelope. This provides a level of collision avoidance and protection to equipment, but is not fully autonomous. It relies on the operator's intelligence to define the envelope. As such, it is a transition capability between teleoperation and total autonomy, combining human intelligence and machine control.

Overall, the main advantage of ManGAP is the ability to provide robotic control and collision avoidance without any additional sensors (and their associated cabling and control hardware). It simply uses the video data that is already present in any telepresence system.

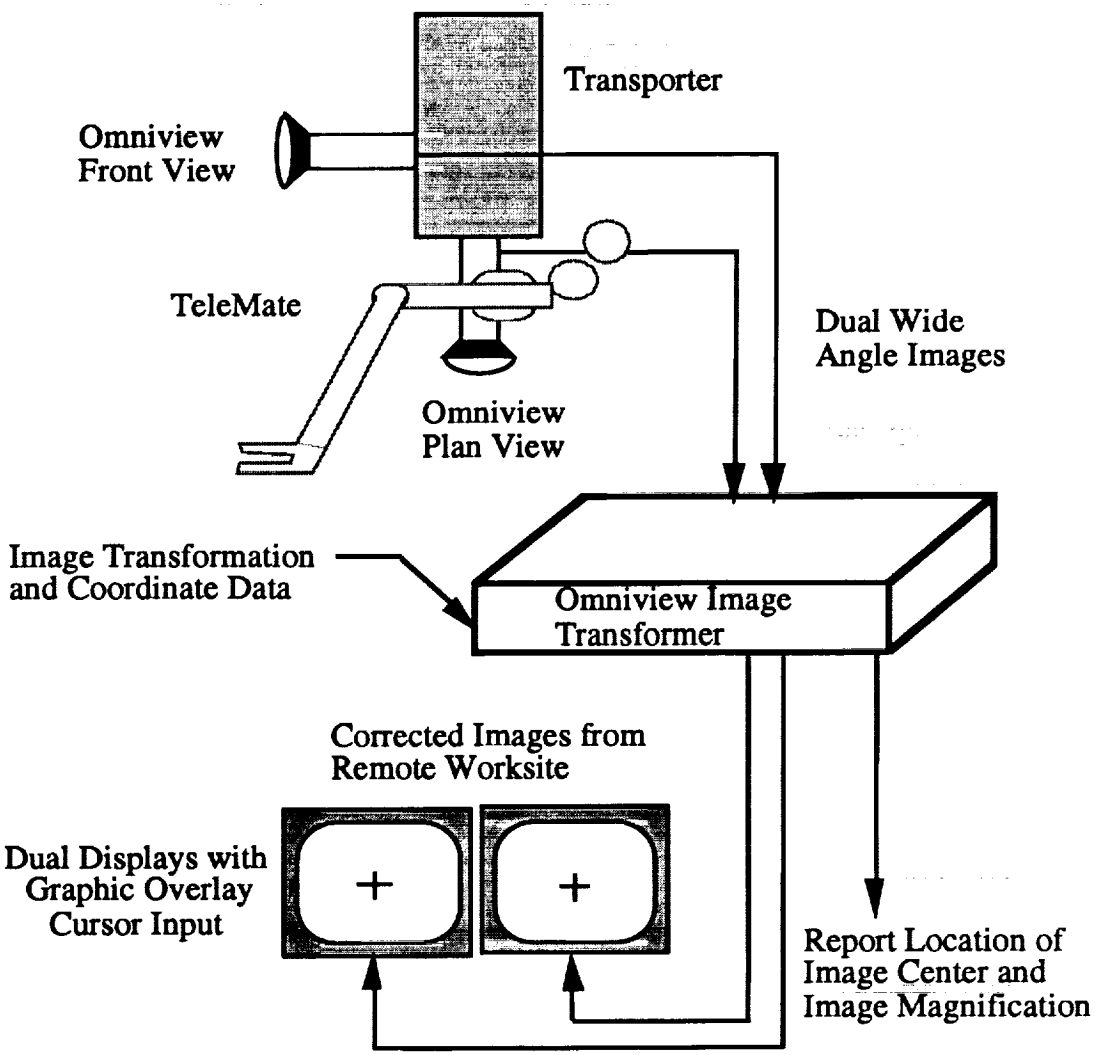


Figure 4 - Manipulator guidance and protection system hardware block diagram.

SUMMARY

Omniview™'s unique capabilities provide significant advantages in teleoperation and virtual environments. The feasibility of telerobot position control and collision avoidance using only video data promises to simplify telerobotic implementations by reducing sensors, cabling and computational requirements. It can also form the basis for an effective compensation of time-delayed video in teleoperations. The real-time demonstration of video manipulation yields convincing proof of virtual motion. This can improve the efficiency of teleoperations as well as provide alternatives to predictive graphical models.

REFERENCE

[1] Zimmermann, S.D. and Martin, H.L., "A Remote Video Pan/Tilt/Magnify/Rotate System with No Moving Parts", 1992 ANS/ENS International Meeting, Nov. 1992.

Session R6: MANIPULATORS AND END EFFECTORS

Session Chair: Mr. Charlie Price

N94-34032

DEXTEROUS END EFFECTOR FLIGHT DEMONSTRATION

Edward L. Carter - MC/C106
Lockheed Engineering and Sciences Company
2400 NASA Rd. 1
Houston, TX 77258

and

Leo G. Monford* - MC/ER411
Lyndon B. Johnson Space Center
2101 NASA Rd. 1
Houston, TX

Abstract

The Dexterous End Effector Flight Experiment is a flight demonstration of newly developed equipment and methods which make for more dexterous manipulation of robotic arms.

The following concepts are to be demonstrated:

The Force Torque Sensor is a six axis load cell located at the end of the RMS which displays load data to the operator on the orbiter CCTV monitor.

TRAC is a target system which provides six axis positional information to the operator. It has the characteristic of having high sensitivity to attitude misalignment while being flat.

AUTO-TRAC is a variation of TRAC in which a computer analyzes a target, displays translational and attitude misalignment information, and provides cues to the operator for corrective inputs.

The Magnetic End Effector is a fault tolerant end effector which grapples payloads using magnetic attraction.

The Carrier Latch Assembly is a fault tolerant payload carrier, which uses mechanical latches and/or magnetic attraction to hold small payloads during launch/landing and to release payloads as desired.

The flight experiment goals and objectives are explained. The experiment equipment is described, and the tasks to be performed during the demonstration are discussed.

DEXTEROUS END EFFECTOR FLIGHT DEMONSTRATION

1.0 INTRODUCTION

The DEE project is a flight technology demonstration. IT is managed by the Automation and Robotics Division of the NASA Johnson Space Center (JSC). The project, with its precursors, began in 1985 as an effort to develop a force torque sensor (FTS) for the Shuttle Remote Manipulator System (RMS). It is currently a flight demonstration with four new technology products to display, and with the additional objective of collecting RMS performance data. DEE is manifested to fly on STS-62 in February of 1994. After a brief overview of the project goals and background, this paper will focus on the flight experiment.

1.1 PROJECT GOALS

The goals of the DEE project are to demonstrate new technology, to gain experience with the hardware and software developed, and to evaluate the benefit to the operator/RMS in performing space operations. The new concepts and hardware are: (1) Force Torque Sensor (FTS); (2) Magnetic End Effector (MEE); (3) Target and Reflective Alignment Concept (TRAC) which can be used manually or automatically; and (4) carrier latch assembly (CLA).

1.2 PROJECT BACKGROUND

The magnetic end effector (MEE) was conceived and developed at JSC. Since the first tests of the MEE/FTS prototype in September 1987, the DEE project has operated frequently at the Manipulator Development Facility (MDF). The Targeting and Reflective Alignment Concept (TRAC) system was developed shortly after the MEE prototype was first used and has been employed in almost all of the MDF operations with the MEE and FTS. Each time a new procedure was developed or a new feature was added to the MEE or to the TRAC system, the change was checked out and demonstrated. These demonstrations have been used to prove new capabilities of the tools, as well as to familiarize interested people with the work being done.

1.3 OBJECTIVES OF THE FLIGHT EXPERIMENT

The detail objectives of the flight experiment are to demonstrate and evaluate the benefits to RMS operators and the task capability of the following:

(a) Use of the Force Torque Sensor to minimize loads on the RMS,
(b) RMS Constrained Control Resolution with the FTS output used for load control,
(c) Generic constrained motion tasks with RMS,
(d) RMS Unconstrained Control Resolution using TRAC for measurements,
(e) Magnetic End Effector enhanced grappling ability and fault tolerance,
(f) Determine capture envelope of the Magnetic End Effector,
(g) TRAC flat mirror target system for improved alignment ability,
(h) Performance data base for RMS,
(i) Force torque sensor using laptop computer with TRAC display,
(j) Electronic cross hairs on orbiter CCTV monitor
(k) AUTO-TRAC computer generated alignment cues
(l) The value of right angle TV camera,
(m) The use of a fault tolerant latch assembly (secondary release capability not required),
(n) Collect arm control data for analysis,
(o) Dynamics of RMS structure and joint drives,
(p) RMS control hypothesis and control logic,

2.0 GENERAL DESCRIPTION OF THE FLIGHT EXPERIMENT

2.1 OVERVIEW

The demonstration will be described from a systems approach, as to the physical arrangement, and from an operations viewpoint.

The DEE is intended to demonstrate five new technologies: the FTS, the MEE, the TRAC, AUTO-TRAC, and the CLA. In the demonstration of these five systems all 16 of the objectives listed above will be accomplished.

In addition to the five technologies there is a support structure and a system of generic tasks which support the demonstration of the five main systems.

The equipment for the five technologies are physically integrated and/or split up by the hardware arrangement.

2.1.1 SYSTEM LEVEL DESCRIPTION

2.1.1.1 FORCE TORQUE SENSOR

The FTS is a load cell which provides six-axis force data to the RMS operator. The FTS is in two parts. The Data Collection Assembly (DCA) is in the payload bay (on the MAT), and the Display Electronics Assembly (DEA) is in the aft flight deck (AFD). These are connected by the RMS special purpose end effector (SPEE) cable.

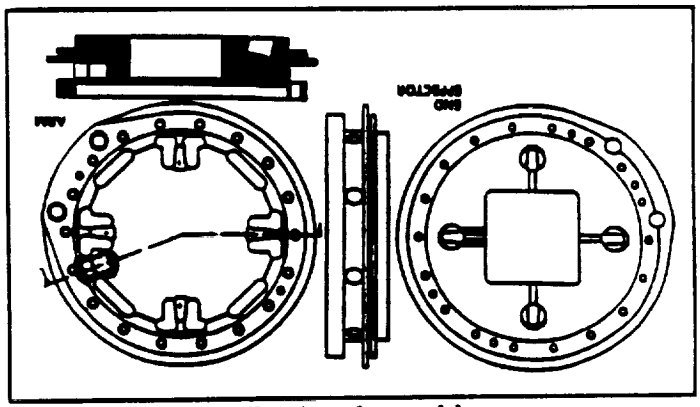


Figure 1 - Data Collection Assembly

The DCA (see figure 1) provides power to 32 strain gages, and, on command from the DEA, it collects the bridge outputs, digitizes the outputs, resolves the outputs into six axis loads (in engineering units), serializes the data into an RS422 bus format and transmits the data to the DEA.

The DEA consists of the SC-1D computer and the video graphics generator (VGG). The DEA performs scaling and point of resolution translations on the signals from the DCA and converts the data into a video display which is viewed on the orbiter CCTV monitor. The monitor display of the VGG output is shown in figure 2. The DEA also receives commands for scale and point of resolution from the Payload General Support Computer (PGSC) and outputs data to the PGSC for recording on floppy disks.

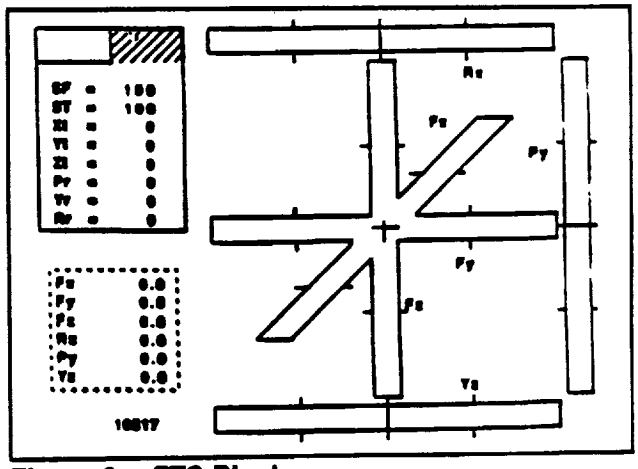


Figure 2 - FTS Display

2.1.1.2 MAGNETIC END EFFECTOR

The MEE is a system which provides for two fault tolerant grappling of payloads by magnetic attraction. A structural housing contains the various MEE components (see figure 3). The primary components are two magnet assemblies, two TV

cameras, backup batteries, and the alignment pins. In addition, there are switches, indicators, camera lights, control circuit boards, and a TV interface device. The MEE produces a magnetic attractive force of 3200 pounds.

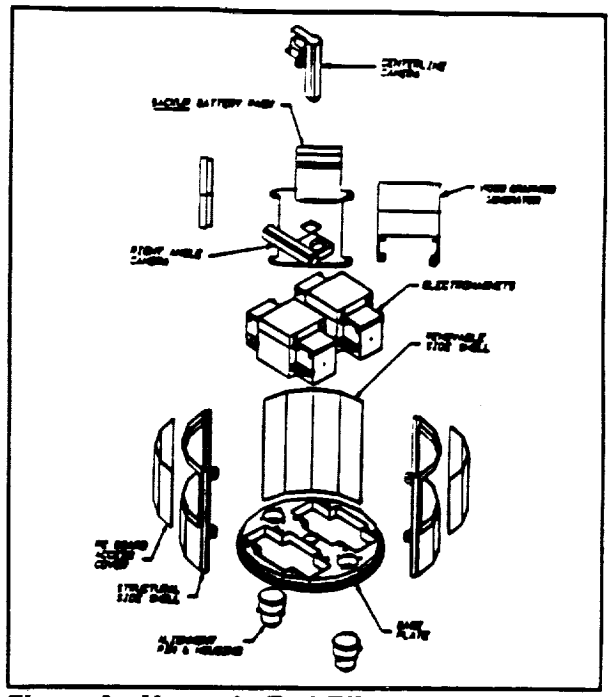


Figure 3 - Magnetic End Effector

2.1.1.2.1 ELECTROMAGNETS.

The two magnets are U-shaped, with three separate coils on each. One is a high powered pull-in coil which produces an appreciable attractive force with a large air gap, and which is automatically switched off by the preload indication system after grapple has been achieved. The other two are holding coils and are identical, with each producing sufficient magnetization to saturate the core and thus develop the full rated holding performance of the MEE. One of the holding coils on each magnet is connected to separate controls and power sources, while the other two holding coils (one on each magnet) are connected to a third power source for two fault tolerant operation. The magnets are arranged with the pole faces within a 7.0-in. square footprint; they are independently mounted on a spring suspension systems in such a way that the poles move slightly toward the grapple fixture during the grappling process. This motion is detected by optical switches as an indication of preload. The use of the springs does not reduce the attractive force, but rather ensures that a preload exists across the grapple interface.

2.1.1.2.2 TV Cameras

Two TV cameras are mounted in the MEE. One is on the MEE centerline and the other normal to the centerline. The cameras are used only for targeting; thus they are preset to a fixed focus distance, and the lens apertures are also preset. Supplementary incandescent lighting is provided for the centerline camera during close targeting. Only one camera output can be utilized at a time.

2.1.1.2.3 Battery Backup

A failure of the RMS exists whereby the electrical connector at the EFGF can become disconnected, thus disconnecting the MEE from all Shuttle power and from all controls. The MEE must not release a grappled payload because of this failure. To accommodate this possible situation, the MEE is equipped with two 18-volt battery backup systems, each of which powers one of the magnet holding coils. The MEE can therefore survive loss of connection and still be one fault tolerant for inadvertent release of a grappled payload.

2.1.1.2.4 ALIGNMENT PINS

The MEE is designed with two spring-loaded alignment pins which ensure accurate alignment and provide increased capability for shear and torsion loads. Optical switches detect the fully out position of the pins.

2.1.1.3 TARGETING AND REFLECTIVE ALIGNMENT CONCEPT

The TRAC system uses a TV camera viewing its own image in a mirror target to achieve alignment in all six axes. TRAC consists of a TV camera, a TV monitor with alignment marks, and a mirror target with cross hairs (see figure 4). Mirror targets are located on objects to be grappled and areas to be targeted. The system can be utilized with the centerline camera, the right-angle camera, or the RMS wrist camera.

In use the target is aligned in all six axes when the reflected image of the camera is centered on the mirror cross hairs, both are centered on the monitor, and the camera image size matches the alignment marks. Translation errors are indicated by the cross hairs appearing off the monitor center and by the size of the camera image being too large or too small. Attitude errors are indicated by the camera image being misaligned to the cross hairs and by the rotational misalignment of the cross hairs to the monitor. The attitude cues are thus separate from the translation cues, and this fact improves operator performance.

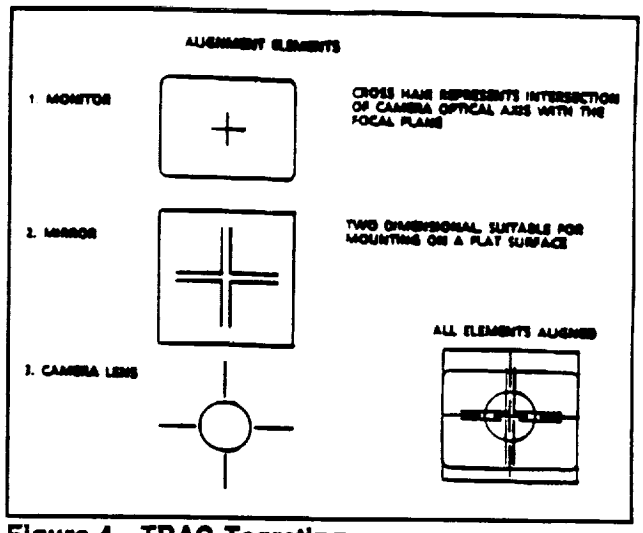


Figure 4 - TRAC Tagreting

2.1.1.4 AUTO-TRAC

AUTO-TRAC is an advanced development of TRAC in which the TV image is processed by a computer to generate alignment errors or operator cues. For AUTO-TRAC five retro-reflectors are mounted on the target mirror (on the middle of each side and on one corner), and an array of light emitting diodes (LED's) are mounted close to the camera lens. Thus when the LED's are emitting and the TV camera is aligned with the target the camera image includes the five retro-reflectors with the direct mirror reflection of the LED's in the center of the pattern. The LED's are made to flash so that in some video frames the LED's are off, but in other frames one or more LED's are on. A frame of video with the LED's off is processed with an adjacent frame of video with an LED on to produce a pseudo-frame of video in which only the LED reflections are present. The processing eliminates the effect of ambient light and simplifies the scene. The pseudo-frame is analyzed for alignment errors.

Control of which LED in the array is on in a given frame allows the direct mirror reflection to be differentiated from the retro-reflector images. Pitch and yaw errors are derived from the amount and direction that the mirror reflection of the LED's is off center relative to the retro-reflector pattern. Roll error is derived from the rotation of the retro-reflector pattern in the video image. Translation errors are derived from conventional stadiametric methods. Singularity ambiguities present in systems using only stadiametric methods are therefore eliminated.

AUTO-TRAC uses a TV camera mounted in the payload bay near the keel and a target mounted on the MAT.

2.1.1.5 CARRIER LATCH ASSEMBLY

The CLA is a small payload carrier which is designed to release a payload to the RMS during on orbit operations. It uses a combination of electro-magnetic holding and electro-mechanical latch pawls to meet the requirements of safety and mission success. The magnets have redundant features identical to those described above for the MEE, except there are no batteries.

In operation, the payload is held mechanically by two sets of independent latch pawls during launch and landing. When release is required, the payload is first grappled magnetically which unloads the mechanical latch pawls. The mechanical latches are then driven open by redundant drive mechanisms, motors, and controls. Indicators are provided for each critical function. The payload can then be safely grappled by the RMS because there are three ways to interrupt electrical power to each set of magnets.

Stowage of the payload back into the CLA follows the reverse sequence.

2.1.2 HARDWARE DESCRIPTION

The DEE equipment is located in three areas, 1) a computer mounted in the (AFD), 2) a targeting camera mounted on a frame in the payload bay, and 3) a longeron-mounted Experiment Stowage and Activities Plate (see figure 5) (a portion of which is released when grappled by the RMS using the Special Purpose End Effector (SPEE)).

The DEE does not affect the standard configuration of the RMS or any other payload using the RMS.

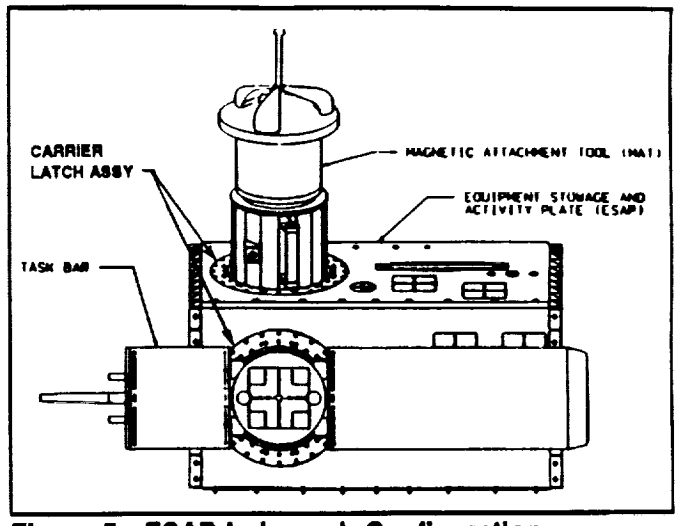


Figure 5 - ESAP In Launch Configuration

2.1.2.1 AFT FLIGHT DECK INSTALLATION

The installation in the AFD consists of parts of the DEA, one-half of a standard switch panel, interconnecting cables, and some standard Orbiter equipment.

2.1.2.1.1 DISPLAY ELECTRONICS ASSEMBLY

The DEA is installed in position L11-Outboard. It provides three switch/circuit breakers and connectors for video and an RS232 port on its front panel.

2.1.2.1.2 STANDARD SWITCH PANEL

The SSP (one-half) provides all of the switches for control of DEE.

2.1.2.2 PAYLOAD BAY TARGETING CAMERA

The targeting camera is a modified commercial TV camera which is equipped with an array of LED's around the lens. It is mounted with a video converter on a small housing on the frame at x=807 and between y=24 and y=34. The converter also provides regulated power and controls the flashing of the LED's. The camera is connected to the standard orbiter keel camera cable.

2.1.2.3 EXPERIMENT STOWAGE AND ACTIVITY PLATE (ESAP)

The ESAP (figure 5) is the structure which is mounted on a Goddard Get Away Special (GAS) Beam and which supports the MAT and Task Bar during launch and landing via two CLA's. In addition, it provides four sockets and seven TRAC targets, which are used in carrying out the experiment operations. The MAT and Task Bar are released to the RMS during demonstration operations.

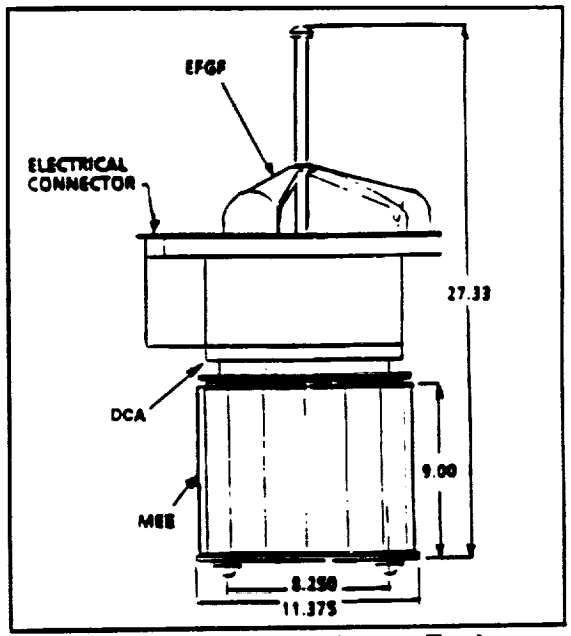


Figure 6 - Magnetic Attachment Tool

2.1.2.3.1 MAGNETIC ATTACHMENT TOOL (MAT)

The MAT (see figure 6) is the assembly which is grappled by the RMS for experiment operation. It is

mounted in the top CLA on the ESAP during launch and landing (see figure 1). The magnetic attachment tool is made up of the MEE, the DCA, and the electrical flight grapple fixture (EFGF). There is also an adaptor between the FTS and the EFGF. The MEE and the DCA hardware are adequately described under 2.1.1.1 and 2.1.1.2.

2.1.2.3.1.1 ELECTRICAL FLIGHT GRAPPLE FIXTURE

The EFGF is a piece of standard STS-provided equipment. For this flight experiment it will be modified by removing a portion of the abutment plate to improve visibility around the EFGF when the TRAC system is used with the RMS wrist TV camera.

2.1.2.3.2 TASK BAR

The task bar, a short panel structure as shown in figure 7, is the device which the MEE magnetically grapples and manipulates during the task operations. One end of the task bar simulates a generic panel, and the other end simulates a module servicing tool (MST).

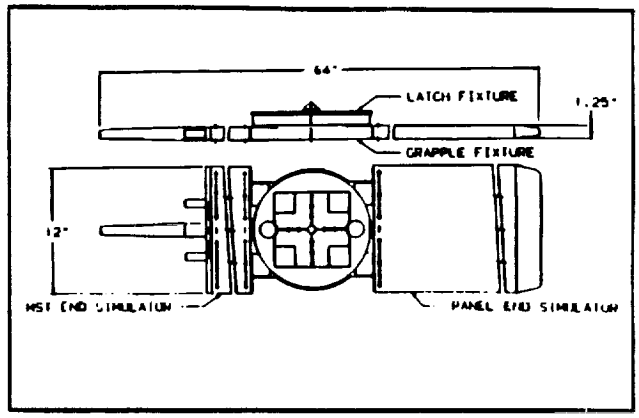


Figure 7 - Task Bar

3.0 EXPERIMENT OPERATION

The task operations for the flight experiment include the following:

a. RMS control resolution tasks
b. Generic constrained motion tasks
c. Magnetic hold down task
d. AUTO-TRAC task

3.1 INITIAL HARDWARE CHECKOUT

The RMS is powered up and uncradled, and the RMS is placed in the vicinity of the MAT. The CLA electromagnets are then energized, and upon holding verification, the mechanical latches are released. The RMS operator then aligns the SEE with the MAT and grapples the MAT. MAT operational capability is now verified. The CLA electro-magnets are turned off and the RMS moves the MAT away from the ESAP. Once the RMS is configured, the experiment tasks begin.

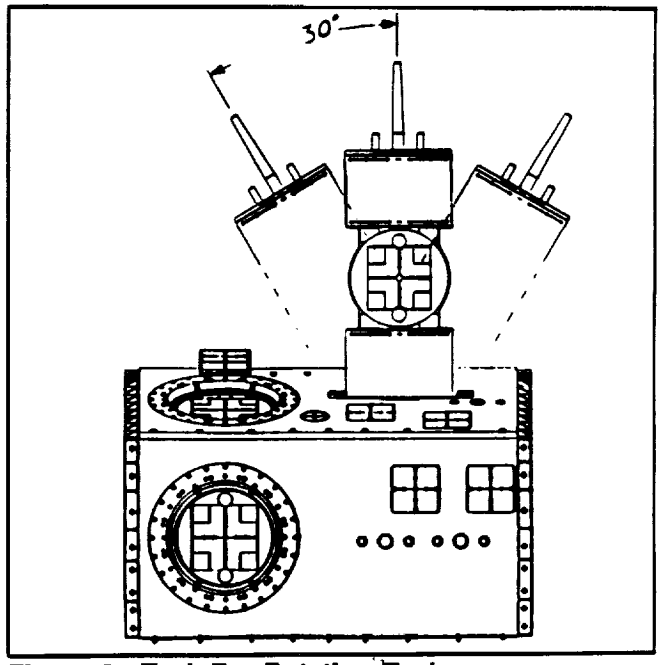


Figure 8 - Task Bar Rotation Task

3.2 RMS CONTROL RESOLUTION TASKS

RMS control resolution is to be determined for unconstrained position alignment control and for constrained force control.

3.2.1 UNCONSTRAINED CONTROL RESOLUTION

The MAT is positioned over a TRAC target, and the operator is asked to align to the target as closely as possible. The errors and the uncommanded RMS motion will be recorded for postflight data analysis.

3.2.2 CONSTRAINED CONTROL RESOLUTION

The MAT is grappled to the task bar while the task bar is in its CLA. The operator is asked to input small forces or to maintain the forces as small as possible. The error and the residual forces will be recorded for postflight data analysis. This data will also be analyzed real time to insure that the control required for the other tasks is within the RMS capability.

3.3 TASK BAR GRAPPLE

Using TRAC for alignment, the MAT is magnetically grappled to the task bar located as shown in figure 8. The task bar is then released from the experiment carrier.

3.4 GENERIC CONSTRAINED MOTION TASKS

3.4.1 PANEL INSERTION AND ROTATION TASK

The RMS is translated to the rotating panel task area. Using the TRAC mirror and MAT right-angle TV camera, the task bar is aligned with the mating slot. With the correct FTS display showing and being monitored, and TRAC alignment maintained as shown by the right-angle view, the task bar is inserted into its mating slot. Full insertion is detected by monitoring the digital readouts on the RMS display and control panel and by observing a stripe on the Task Bar. A roll will be performed (see figure 8) and loading on the task bar will be monitored, up to approximately +/-30°.

3.4.2 MODULE SERVICING TOOL SIMULATION TASK

Simulation of the MST operations begins with the MAT grapple of the task bar and the subsequent wrist roll of the task bar to the vertical position. Using the corresponding TRAC target, the task bar probe is aligned with the receptacle and inserted into the receptacle while forces and torques exerted on the task bar are minimized as before. Several methods of insertion may be examined as time permits.

3.5 MAGNETIC HOLD DOWN TASK

Between the panel insertion task and the MST simulation task, the task bar is temporarily restowed on its latch assembly. The MAT then releases the task bar, leaving it on the latch assembly with only the electromagnets holding the task bar. This demonstrates the magnetic hold down task. Next, the MAT is rolled 180° and regrappled to the task bar.

3.6 AUTO-TRAC DEMONSTRTION

The MAT will be positioned so that the AUTO-TRAC target will be aligned in the view from the targeting camera. The position will be recorded from the RMS joint angles. The MAT then will be moved to a misaligned position. Next the RMS will be commanded to return to the recorded position using the auto sequence mode of operation. This will be repeated several times from different conditions of misalignment. The residual alignment errors will be recorded for post flight analysis.

4.0 CONCLUSION

The DEE flight demonstration has the potential for bringing five new developments into the realm of technology for use in space. With the completion of the STS-62 demonstration the concepts will be proved, and the hardware designs will be available for other users. Some of the demonstration hardware may be available for other flights. The use of these concepts and/or hardware will improve the efficiency, lower the cost, improve safety, and even allow totally new concepts of how men work in space.

N94-34033

Undersea Applications of Dexterous Robotics

Mark M. Gittleman
Oceaneering Space Systems
Houston, TX

The author of this paper will examine the evolution and application of dexterous robotics in the undersea energy production industry and how this mature technology has affected planned SSF dexterous robotic tasks.

Undersea telerobotics, or Remotely Operated Vehicles (ROVs), have evolved in design and use since the mid-1970s. Originally developed to replace commercial divers for both planned and unplanned tasks, they are now most commonly used to perform planned robotic tasks in all phases of assembly, inspection, and maintenance of undersea structures and installations. To accomplish these tasks, the worksites, the tasks themselves, and the tools are now engineered with both the telerobot's and the diver's capabilities in mind. In many cases, this planning has permitted a reduction in telerobot system complexity and cost.

The philosophies and design practices that have resulted in the successful incorporation of telerobots into the highly competitive and cost conscious offshore energy production industry have been largely ignored in the space community. The author of this paper will explore cases where these philosophies have been adopted or may be successfully adopted in the near future.

Undersea Applications of Dexterous Robotics

Mark M. Gittleman

Oceaneering Space Systems

Outline

I Introduction

II Operational and Design Philosophy

III ROV and Task Evolution

IV Current Practices and Designs

V Effects on SSF

VI Implications for Future Missions and Facilities

Introduction

THE UNDERSEA OIL AND GAS PRODUCTION INDUSTRY HAS BEEN USING DEXTEROUS TELEROBOTIC SYSTEMS SUCCESSFULLY FOR OVER 15 YEARS

- It is a highly competitive industry requiring cost effectiveness at all times.
- Several competing types of Work Systems are available
  - Hyperbaric divers to approximately 1,000 feet of seawater (FSW)
  - Atmospheric dividing systems (ADSs) to approximately 2,200 FSW
  - Manned submersibles to approximately 3,000 FSW
  - Remotely Operated Vehicles (ROVs) to 20,000 FSW
  - Most can compete in the 300 FSW to 1,000 FSW range on certain tasks

Introduction (cont'd)

- Within each type of work system are numerous variations with different capabilities. For example:
  - ROV capabilities range from simple inspection telerobots (flying eyeballs) to highly complex multipurpose work systems
  - ROV manipulators range from 3 to 6 DOF and from simple, open loop control systems to spatially correspondent bilateral force reflection

Operational and Design Philosophy

TASK REQUIREMENTS MUST DRIVE WORK SYSTEM AND TOOL SELECTION AND DESIGN. TOOLS, INTERFACES, AND WORK SYSTEMS SHOULD BE DEVELOPED AS INTEGRAL PARTS OF A NEW FACILITY

- This means that new work systems are not developed as technology demonstrations.
  - For almost 15 years, ROV manipulators steadily became less complex
  - Worksites and tools were developed that allow simpler, rate controlled arms to deliver "smart" tools
  - Systems engineering is the key to success
  - Tasks should be designed to the midrange of a work system's capabilities
- Selection of a work system to perform a task is driven by:
  - Safety considerations
  - Cost effectiveness
  - Availability
  - Operational capability

Operational and Design Philosophy (cont'd)

OPERATIONS AND LIFE CYCLE COSTS CAN BE OPTIMIZED BY BALANCING THE COMPLEXITIES OF THE TOOLS AND PROCEDURES WITH THE CAPABILITIES OF THE WORK SYSTEMS

- This means understanding and defining the task requirements *before* selecting/designing work systems and tools.
  - Selecting/designing work systems and tools first can result in over complicating the hardware and procedures
- The best way to integrate manned and robotic resources is through tools and interfaces.
  - They are used to bridge the gap between work system capabilities and task requirements
  - Tools are the cheapest and easiest component of the system to exchange, upgrade, lose, or break
  - Common tool use techniques can simplify training and reduce long-term costs

ROV and Task Evolution

- Early ROVs were called upon to perform tasks not designed for remote intervention
  - Frequently, these tasks were not designed for *any* intervention (i.e., "diverless systems")
  - Early tasks required highly dexterous, costly, force reflecting manipulators that were intended to replace divers
  - As a result of inappropriate designs and overselling ROV capabilities, there was a backlash that resulted in a several year delay in the acceptance of ROVs
- As a result of the lessons learned in the early years, ROV designers and operators scaled back claims and learned to walk before they ran
  - Result was inspection ROVs and simple dedicated work ROVs for the 1970s and early 1980s

ROV and Task Evolution (cont'd)

- Starting in the early 1980s, Systems Engineering allowed complicated, expensive, and sometimes unreliable manipulators to be replaced by simple manipulators and end effectors, properly design worksites, and smart tools
  - ROV designers took the time to examine task and customer requirements and designed accordingly
  - Tasks evolved from inspection to drilling support to complex assembly and maintenance tasks
  - ROVs gradually gained general acceptance and now account for roughly 25% of undersea intervention
- One of the critical lessons learned is that designing for an ROV generally results in a simpler operation for the diver
  - The result of worksites designed for both ROVs and divers and/or ADS is increased operational flexibility and reduced costs

ROV and Task Evolution (cont'd)

- Facility compatibility with multiple work systems (ROVs, divers, and/or Atmospheric Diving Systems) can be a contractual requirement
  - Requires a systems engineering approach and an understanding of each system's capabilities
  - Results in standard interface(s) for multiple tasks
  - Results in common tooling front ends with back end interfaces designed for the appropriate work system

Current Practices and Designs

- Oceaneering International owns and operates a fleet of over 65 ROVs. Approximately 45 of these are considered "large work vehicles" and include manipulators of various types
  - The majority of these ROVs use simple, reliable, open loop rate controlled hydraulic manipulators with generic grippers
  - The use of these manipulators is now possible because the typical ROV worksite and task are designed for ROV intervention
  - The new advanced work systems may include force reflecting manipulators and digital control
  - Force reflection is generally considered not cost competitive for planned tasks
- These systems are extremely reliable. In the past 7-year period in U.S. waters, Oceaneering's uptime ratio for 8,349 dives (32,028 hrs) was 98% for heavy work ROVs.

Current Practices and Designs (cont'd)

- Advanced ROVs may now include:
  - A large array of sensors and positioning devices
  - Onboard self diagnosis and system health sensors
  - Vision systems comprised of still and video cameras
  - Any of a large number of tools and tool systems
  - Two manipulators, 6 DOF, with generic grippers
  - Spatially correspondent with or without force reflection
  - Digital and analog systems
  - Fiber optic umbilicals
- Typical tasks include:
  - Inspection (visual and NDE) and monitoring
  - Diver support (worksite assessment, diver monitoring, tool retrieval, hydraulic power source, etc.)
  - Construction, assembly, and maintenance
  - Valve actuation
  - Quick response/rapid deployment tasks

Current Practices and Designs

- Typical tools include:
  - Grinding, cutting, and cleaning tools
  - Inspection tools (visual and NDE)
  - Special rigging tools
  - Pipe and cable burial tools
  - Valve override and valve actuation tools
- Some recent ROVs have eliminated manipulators completely. Instead, docking cones with integrated tools are used

Current Practices and Designs (cont'd)

- Advanced ROVs are becoming both more complex and more capable *BUT* no ROV can compete unless it is cost effective
  - The best performance still comes from ROVs performing tasks that are designed for remote intervention *and* are operated by resourceful, experienced people
  - Unplanned tasks benefit from more capable systems but always require the best operators available
- If strong, highly dexterous manipulators and end effectors ever also become inexpensive and reliable, that could create a paradigm shift toward highly dexterous systems that would be much closer to "diver replacements" than current ROVs

Effects on SSF

- Some of the design and operational practices developed for subsea telerobots have been adapted by the space community
  - Robotic Systems Integration Standards (RSIS, SSP 30550) was originally based on subsea lessons learned
  - Many of the interfaces for SSF have been standardized, one is based on typical undersea interface geometrics
  - A limited number of tools have been accepted
- Many of the practices that have resulted in the successful integration of telerobots into the undersea oil and gas industry have been largely or partly ignored by the space community
  - Telerobotic systems are often seen as technology development projects
  - The use of tools to bridge the gap between telerobotic system capabilities and task requirements is often resisted
  - Design for telerobotics is still commonly seen as an add-on to the EVA design
  - Many telerobotic tasks for SSF are designed at the extreme edge of the telerobot's capability

Implications for Future Missions and Facilities

- An integrated systems engineering approach treating each asset as a work system is needed to successfully use telerobotics in space
- Dexterous telerobotics will be accepted as a viable tool in space operations only after they have been successfully demonstrated in an operational environment; e.g., multiple flight experiments ("sea trials")
- Dexterous telerobotics will be successful in space operations when the community accepts the concept of up-front systems engineering that permits the rational development of work systems designed to meet the requirements of the mission
  - Telerobotic systems designed independently of the mission mainstream will be resisted, resented, and probably unsuccessful
- Dexterous telerobots and robots could by highly successful in future missions if the lessons learned over the past 15 years in the undersea environment are used as the basis for further development

N94-34034

Robotic Technologies of the Flight Telerobotic Servicer (FTS) Including Fault Tolerance

John T. Chladek
NASA - Johnson Space Center
Houston, TX 77058

William M. Craver
Lockheed Engineering and Sciences Co.
2400 Nasa Road 1, Houston, TX 77058

Anticipated Mission Tasks

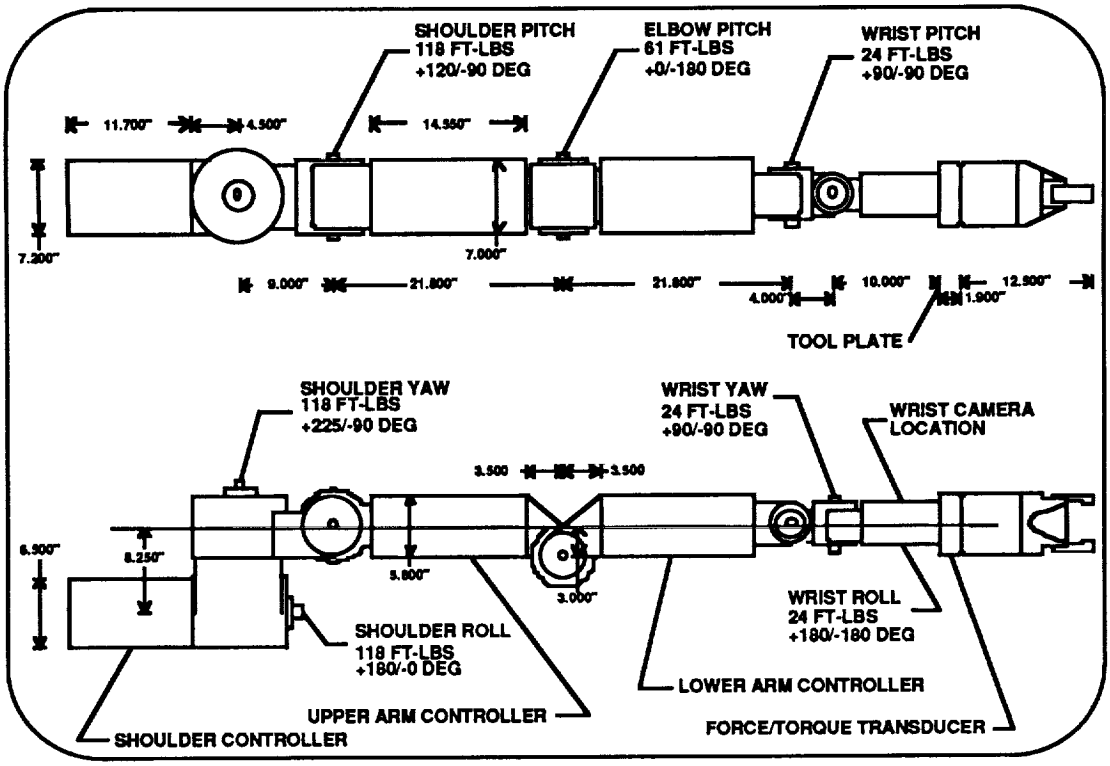
The original FTS concept for Space Station Freedom (SSF) was to provide telerobotic assistance to enhance crew activity and safety and to reduce crew EVA (Extra Vehicular Activity) activity. The first flight of the FTS manipulator systems would demonstrate several candidate tasks and would verify manipulator performance parameters. These first flight tasks included unlocking a SSF Truss Joint, mating/demating a fluid coupling, contact following of a contour board, demonstrating peg-in-hole assembly, and grasping and moving a mass. Future tasks foreseen for the FTS system included ORU (Orbit Replaceable Unit) change-out, Hubble Space Telescope Servicing, Gamma Ray Observatory refueling, and several in-situ SSF servicing and maintenance tasks. Operation of the FTS was planned to evolve from teleoperation to fully autonomous execution of many tasks.

This wide range of mission tasks combined with the desire to evolve toward full autonomy forced several requirements which may seem extremely demanding to the telerobotics community. The FTS requirements appear to have been created to accommodate the open-ended evolution plan such that operational evolution would not be impeded by function limitations. A recommendation arising from the FTS program to remedy the possible impacts from such ambitious requirements is to analyze candidate robotic tasks. Based on these task analyses, weigh operational impacts against development impacts prior to requirements definition. Many of the FTS requirements discussed in the following sections greatly influenced the development cost and schedule of the FTS manipulator. The FTS manipulator has been assembled at Martin Marietta and is currently in testing. Successful component tests indicate a manipulator which achieves unprecedented performance specifications.

Functional Requirements

The functional requirements of the manipulator involve environmental, performance, safety, and resource effects. Many of these requirements are driven by the space environment, such as operation in thermal extremes, the need for safety, and limited resource availability (weight and power). Most of these requirements, however, focus on the manipulator and component functions to insure superior performance and ability to upgrade (evolution toward autonomy).

The primary robotic function of the FTS manipulator is that it move or manipulator objects in zero-gravity. Because interchangeable end-effectors were being considered, the manipulator requirements specify the tool-plate as the point of reference. The tool plate is the attachment point for the wrist force/torque sensor. A manipulated object's mass may be as high as 37 slugs (1200 lb.) with the manipulator able to move masses less that 2.8 slugs (90 lb.) at velocities of 6 inch/second. Unloaded tool plate velocity will be at least 24 inch/second. Accuracy of tool-plate positioning relative to the manipulator base frame must be within 1 inch and ± 3 degrees. The manipulator must be able to resolve tool-plate incremental motion within 0.001 inch and 0.01 degrees. Additionally, repeatability must be within 0.005 inch and ± 0.05 degrees with respect to the manipulator base frame. To perform useful work, the FTS manipulator was required to provide 20 pounds force and 20 foot-pounds torque output at the tool plate in any direction and in any manipulator configuration. These output force and positioning requirements were to be utilized with several control schemes including joint control, Cartesian control, and impedance control.



The FTS Manipulator

To operate in space, the FTS manipulator had to meet the shuttle safety requirements as well as the environmental extremes. The safety requirements, as discussed elsewhere in this paper, ensure Orbiter and crew safety through fault tolerance requirements. Safety is cited by Shattuck and Lowrie (1992) as "the single largest factor driving the system design." Safety and fault tolerance requirements resulted in monitoring of joint and Cartesian data, in checking of loop times to ensure proper functioning, in cross-strapping along communication paths, and in addition of a hardwire control capability as a backup operational mode. Orbiter launch and landing impart vibration into the system which requires structural analysis and testing. Electromagnetic interference (EMI) must be limited both from invading and from exiting the manipulator systems. However, the most demanding aspect of the space environment from the FTS designer's view is the thermal vacuum of space. Operation in a hard vacuum (10-5 torr) and over temperatures from -50°C to 95°C forces innovative designs, careful material selection, and extensive analysis.

Another consequence of the space environment is operation in zero-gravity. Designing the manipulator for a zero-g environment impacts structural, electromechanical, and electrical power considerations and well as the control system design. Because weight is a premium in space, motors are chosen to provide torques for zero-g operation. This saves significant weight and electrical power when compared to motors chosen for ground-based operation. Smaller motors also benefit the thermal control system. The structure must also be lightweight, which increases flexibility and lowers structural bending mode frequencies. While being lightweight and more flexible, space manipulators are expected to handle payloads more massive than the manipulator. This expectation is far different from terrestrial manipulators which usually handle payloads 1/10 their weight. To maintain stability and performance, a 10:1 ratio is maintained between the first bending mode and the control bandwidth. This ratio precludes use of high bandwidth PID servos used in more massive, terrestrial manipulators. To address the stability and performance issues in the FTS manipulator, the structure was designed for stiffness (12 Hz first bending mode) and the manipulator control has a 1.2 Hz bandwidth, an inertia decoupler, and joint-level torque, position, and velocity servo loops.

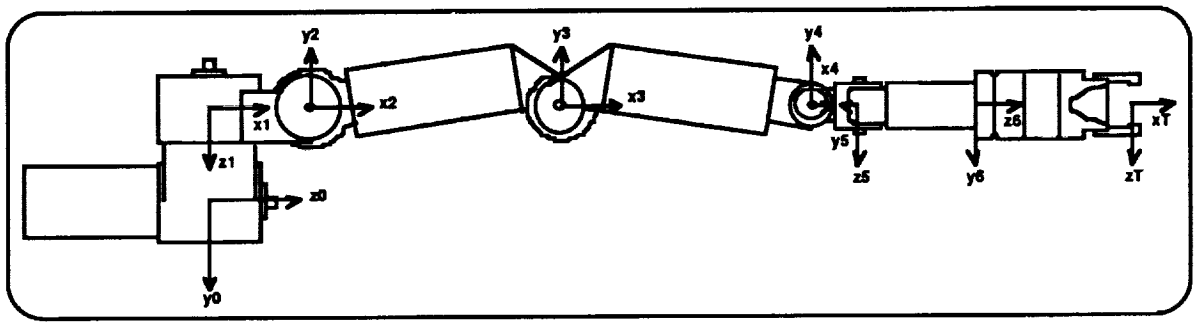
Manipulator Design and Technologies

Beyond safety, FTS manipulator design was driven by the thermal environment and the

positioning performance specifications. Of course, each manipulator subsystem was influenced by additional constraints and specifications. The following paragraphs describe the manipulator subsystem designs and technologies developed by Martin Marietta and its subcontractors to meet the FTS requirements. Manipulator subsystems discussed include manipulator kinematic design, link structure, actuators, control systems, and the end-of-arm tooling.

Manipulator Kinematics

A 7-DOF (degree-of-freedom) R-Y-P-P-P-Y-R design is used with the first joint (shoulder roll) utilized for task-dependent configuration optimization. The outer 6 joints are actively controlled for coordinated output motion. The kinematic design has few joint offsets and 90° twist angles to simplify the kinematics. The 6-DOF kinematic arrangement, with three adjacent pitch joints, provides a closed-form inverse kinematic solution with few singularities within the manipulator workspace. The singularities which occur when the wrist roll or wrist yaw align with the shoulder yaw are beyond the usual workspace of the manipulator. Other singularities occurring at joint limits and when the elbow passes over the "home" position, shown below, are eliminated with mechanical and software joint travel limits. The 3 inch displacement of the elbow joint is to allow the arm to fold back on itself for a greater workspace.



FTS Manipulator - "Home" Position

Link Structure

The manipulator links provide structural support as well as joint controller electronics packaging and thermal control. Packaging and thermal control determined link sizes while fracture and stiffness considerations drove the structural design of the links. A stiffness requirement of 1,000,000 pounds/foot and 1,000,000 foot-pounds/radian resulted in a smallest structural safety margin which exceeds 14, far greater than Shuttle requirement for a 1.4 factor of safety. Easy access to electronics is through side plates on the links. To avoid the cost and complication of active cooling, radiation is the primary thermal path. The controller boards sit in slots within the links which provide conduction paths to the link structure for radiation to the environment. The link designs use material coatings, mounting, and Kapton/Inconel film heaters to maintain thermal control.

Actuators

The joint actuator designs, developed by Martin Marietta and Schaeffer Magnetics, were also driven by positioning, performance, and thermal demands. These high-performance, zero backlash actuators each house a DC-motor, an harmonic drive transmission, an output torque sensor, an output position sensor, a fail-safe brake, hard-stops, and internally routed cabling. The design achieves considerable commonality between actuators. Three sizes are used - one for the 3 shoulder joints, one elbow joint, and one for the 3 wrist joints.

The DC-motors have brushless, delta-wound stators with samarium cobalt rotors. This design offers good thermal properties, low EMI, minimal rotational losses, and linear torque-speed relationships. Motor commutation signals are generated from Hall Effect sensors, a second set of which is installed for redundancy. A secondary set of windings within the stator, driven via an independent electrical path, provides at least 10% rated torque and 0.5 degrees/second joint velocity for operation of a backup mode. This degraded mode of operation, commanded joint-by-joint, satisfies the need for safing the manipulator after failure of a primary system. Fail-safe brakes attached to the motor rotor shaft are spring-loaded so that loss of power engages the brake. These brakes may be released with an EVA release bolt, which when turned 90° releases a cam

on the brake armature.

Harmonic drives provide 100:1 backdrivable gear reduction in a compact volume. The harmonic drives were chosen with HUIC-series cups and S-tooth profile teeth for torsional stiffness and zero backlash. Cup size is determined by joint torsional stiffness requirements. In fact, because of the relative flexibility of the harmonic drive, all other torsion members are considered rigid. Rather than the standard Oldham coupling to the wave generator, a specially designed cylindrical coupler was used to eliminate backlash. Additionally, the output is coupled to a flange around the motor and harmonic drive. This flange, mounted to large duplex bearings provides compactness, rigidity, and an efficient load path the output link.

An analog torque loop is implemented in the joint servos to accommodate the non-linear and high-frequency affects of the harmonic drives. Sensor values to the torque loop come from an output torque sensor embedded on the harmonic drive output flange. Strain gages are mounted to the spokes of the titanium flange. This sensor placement isolates the sensor from structural loads (bending), thus primarily transmitting actuator torque. For effective performance, this analog torque loop operates at 1500 Hz.

Like the manipulator structure, actuator housings and bearings were designed for stiffness and thermal stability. A standard bearing steel, 440C stainless, is used for all bearings. Bearing lubricant is Braycote 601, a liquid lubricant used in space applications. Its very low vapor pressure allows the actuator to not be sealed, but still designed to resist contamination and assembled in a clean room. The motor bearings are deep-groove roller bearings sized for the thrust load of brake engagement and spring pre-loaded to minimize temperature sensitivity. The output bearings are large diameter, duplex-pair, angular contact bearings (face-to-face mounting). These bearings share radial and thrust loads with another duplex-pair on the other side of the actuator. An exception is the wrist roll, which has a single, duplex pair mounted back-to-back for better rigidity against the bending moments of the full cantilever load. Unfortunately, this back-to-back installation has greater sensitivity to assembly misalignments. This sensitivity may contribute to the excessive, uncompensated friction discovered during recent wrist roll torque loop tests.

The actuator housings are aluminum and titanium. Titanium is utilized near bearings. The similar thermal properties of 440C stainless and 6Al-4V titanium minimize temperature effects on bearing pre-loads. These pre-loads were determined as a compromise between stiffness and friction drag. The actuator case was designed for thermal needs. Motor and brake heat is dissipated to the ends or to the casing and then radiated to the environment. Like the links, the actuator design uses thermal isolation, material coatings, and internally mounted film heaters to protect bearings from thermal gradients. These gradients could adversely affect actuator friction and positioning accuracy.

The positioning and incremental motion requirements call for encoder data within an arc-minute at resolutions to 22-bit sensor. To meet this need, inductive encoders were developed specifically for the FTS program by Aerospace Controls Corporation. These encoders have a fine and a coarse track used for incremental and absolute position resolution, respectively. Temperature effects on sensor accuracy were discovered during thermal testing. These errors were stable and repeatable with temperature, and are thus have been corrected in software.

All cabling in the manipulator is internally routed through links and actuators. Each actuator has a cable passageway designed to eliminate twisting of cabling and thus minimizing chafing opportunity. The innovative cabling within these actuators is of Flat Conductor Cables (FCC), manufactured by Tayco, Inc. FCC is used in space applications, but for this application up to 34 layers of laminated cables are used in a single actuator passageway. The cables consist of alternating layers of Kapton, FEP, and photo etched copper conductors with a vapor-deposited copper shield. These cables are to operate from -50°C to 95°C through thousands of cycles. These cables rout serial data, video signals, power, and discrete signals. Acceptance tests of a few cables indicated minor lamination problems apparently due to entrapped water vapor. Investigation of the cable manufacture and test indicated several areas for possible change as well as a method for cable repair. Recent cable tests to 100,000 mechanical cycles over full temperature ranges verified continued cable functionality.

Control Systems

The FTS manipulator control design provides 6-DOF active control over a wide range of payloads as well as impedance control for stable contact. The control algorithms are specified according to the NASREM architecture (NASA/NBS Standard Reference Model for Robotic Systems). NASREM is implemented as a layered architecture with 4 levels: Task, Elemental-Move, Primitive, and Servo. Use of these levels allows operation from teleoperation, the Servo level, advancing to fully autonomous task sequencing, the Task level. Developments to date have focused on the Servo level commands. The Servo level receives Cartesian manipulator commands and transforms them to joint level servo commands. Efforts with the NASREM Primitive level have incorporated point-to-point Cartesian path generation.

The wide payload range specified for the FTS manipulator causes the manipulator joints to experience inertial loads over several orders of magnitude. These loads are induced by the coupling which occurs between joints and affects the trajectory-tracking accuracy of the manipulator. The position controller implemented in the FTS manipulator compensates for these torques with a model-based inertial decoupler. The feed-forward decoupling scheme computes expected inertial torques due to commanded motion and sums this torque with the joint command. The position-dependent inertia matrices used to calculate these torques are computed every 200 ms. This value was chosen as a compromise of accuracy and computational burden.

In addition to the free-space performance requirements, satisfied with the position controller and inertial decoupler, the FTS manipulator must provide stable contact with its impedance control. The impedance controller is position-based, that is, the manipulator and joints are treated as actuators of Cartesian position. Thus, end-effector force measurements are transformed into Cartesian motion commands based on a desired output impedance. This approach was chosen over a torque-based approach because a torque-based approach has instabilities for higher stiffness values and may have difficulty applying large forces to a worksite. Also, a torque-based approach may store energy, resulting in large accelerations when contact is broken. To maintain stability during the transition from free-space motion to contact, a joint velocity feedback term is included for "augmented damping." The resulted lightly damped contact insures stability, but when contact is broken the free-space motion becomes overdamped and sluggish. A feed-forward velocity term is implemented to compensate for this poor free-space response. These control schemes, which increase the complexity of the controller are designed to meet the FTS free-space motion, payload capacity, and contact performance requirements.

An emergency shutdown (ESD) system is embedded in the manipulator control architecture. This system was implemented to provide active control of hazards to meet the payload safety requirement to be two-fault tolerant against catastrophic hazards. The primary hazards in this case are unplanned contact and excessive force generation. The ESD approach is to use 3 control levels to monitor joint and Cartesian positions and velocities, comparing both commands and sensor feedback. A separate ESD bus, which connects the joint, manipulator, and power controllers, is the path by which an ESD is initiated - removing power from the manipulator systems. The first level checks that commanded values are within allowable limits both in the manipulator controller and the joint controllers. The second level monitors safety critical parameters such as position, velocity, and torque with the joint controllers and within the manipulator controller collision avoidance routines. The final level of ESD monitoring is a check of redundant safety critical parameters in the redundant manipulator controller and in independent joint controllers.

In the event of an apparent failure, several possible ESD actions may be automatically initiated. The operator, of course, has a manual ESD to power off the manipulator at any time. If monitored values are elevated but do not pose immediate danger, a soft stop is initiated by the control software. A soft stop commands the manipulator to hold the current position with brakes off (disengaged). An example of a soft stop condition is a Cartesian manipulator command which violates a warning boundary near a known obstacle. A hardware ESD is initiated by any controller when an analog sensor value exceeds its limit - resulting in an ESD notification on the ESD bus. These analog comparisons are being performed at 1500 Hz. A software ESD occurs when a controller CPU detects an out-of -limit condition and signals the power module over the Mil-Std-1553B communication bus. The power module then initiates a combination ESD to power off the manipulator. A combination ESD is detected by software comparisons in the controllers and

initiates a software reset of a hardware limit value to force a hardware ESD. All these ESD paths were analyzed to determine reaction times to various failures such as a joint runaway. Hardware ESD's occur in 11 msec, combination ESD's occur in 30 to 206 msec, and a combination ESD may take up to 4026 msec for an over-temperature condition.

Gripper/End-of-Arm Tooling

The end-of-arm tooling built for the FTS manipulator has a parallel jaw gripper and space for later addition of an end-effector exchange mechanism. The gripper fingers are a cruciform designed for positive contact and retention because the gripper is backdrivable. The gripper fingers ride on a rack and pinion driven by a harmonic drive transmission and a single DC-motor. A pair of fail-safe brakes are installed to provide fault tolerance against inadvertent release. Brake failure or brake command failure results in a brake defaulting to its engaged position. Each of the two brakes can withstand forces greater than expected gripper forces (maximum anticipated load is 30 lb, brake hold is 50 lb.). Gripper forces are measure by a torque sensor and also by motor currents. The concern over inadvertent release also impacted the design the planned task items. These items were instrumented to insure positive grasp. As a final safety measure, the gripper fingers are attached with EVA compatible bolts which may be removed on-orbit to release the gripper.

SAFETY REQUIREMENTS

Robotic Manipulator Systems can provide the capability to perform work and assist humans in space as long as they are safe and reliable. The space based requirements differ significantly from terrestrial based manipulators used in industry and research. In most terrestrial robot implementations, the prime method for dealing with failures is to keep workers out of the robot workspace when active and by accepting the occasional parts damage following a failure due to high volume parts fabrication. This approach is not acceptable for space applications where humans are involved, and the effect impacts the design requirements for space manipulator systems.

Hazards and Controls

All manned space flight systems are assessed for flight hazards their use imposes. From such an assessment the causes of those hazards are determined, and methods to control those hazards are developed. To gain flight acceptance, multiple levels of hazard control must be designed for and verified for assuring the desired level and coverage of controls. In the FTS system development, safe control of hazardous operations forced additional requirements in the design of the manipulator system, its interfaces with the Orbiter and the task elements the FTS was to demonstrate interaction with.

The primary hazards associated with the FTS manipulator operations and the three methods for providing safe control are listed:

A) Unplanned contact or impact during operations
   1) Operator and computer control to not command unplanned contact.
   2) Boundary management software operation.
   3) Redundant boundary management software operation in the safety computer
B) Inadvertent release of hardware
   1) Hardwired enable gripper brake power from independent switch in the aft flight deck
   2) PGSC (Portable General Support Computer: laptop computer) command to release gripper Brake #1
   3) Hand controller switch to release gripper Brake #2
C) Failure to stow for safe Orbiter landing
   1) Normal computer operations (With hardwired control for added reliability)
   2) Jettison via RMS (or EVA if time permits)
   3) EVA operations to stow or jettison
D) Excessive applied gripper force or torque
   1) Force control using gripper force sensor
   2) Current limiting ESD (Emergency shutdown detection)
   3) Redundant current limiting ESD
E) Excessive applied manipulator force or torque
   1) Normal control with active Cartesian load from joint torque command
   2) Cartesian force limiting, using wrist force/torque sensor channel A
   3) Redundant Cartesian force limiting, using wrist force/torque sensor channel B.

Mission Operation To Control Hazards

Primary concerns in the design of space manipulator systems have to do with the effects of system failures on the crew or vehicle. Operational limitations of use are placed on robotic systems that may otherwise be perfectly capable of performing their intended operations. Limitation on use are due to the fact that if a system is performing a task and were to have a failure, the effect of that failure must not prohibit the intended function from being performed in the time frame that that function is critically needed, and any failure must not prohibit any other safety related operations from being carried out during its time of criticality..

For a system to continue operations after a failure, any remaining operability the system might contain must also provide that same capability to make itself safe to the vehicle and crew if it were to suffer a failure. Otherwise that additional level of operability would only be allowed for temporary use to make the task situation safe, remove the robot from the task area, and then stow it in a safe returnable state or eject it so the vehicle can return to Earth. The added operability would not be allowed for continued use to proceed with the intended task, except to make the situation safe. This is the fundamental concept of hazard control for the Orbiter.

FTS Fail Safe Operations

Several FTS configuration descriptions follow below along with design features to address key functions which allow for safe operations. The designs comply with NASA's Orbiter safety policy and requirements of NSTS 1700.7B with interpreted in NSTS 18798A. In several cases, the hardware or software system could not be designed to meet the required levels of fault tolerance without significantly complicating the design or dexterity of the manipulator system. Therefore reductions in compliance with the safety requirements placed operational limitations on the use of the FTS System. The system is considered fail safe; where under any failure the system will not cause a catastrophic hazard, and therefore does not jeopardize the safety of the Orbiter or crew. The FTS system is not fail-operational. Such a system, after any initial failure, could continue normal intended operations since it would still retain the ability to make itself safe after a second failure.

The DTF-1 concept fulfills the first method of hazard control for Orbiter safety using its normal modes of operation. If any of the single points of failure occur, normal operations will cease and an attempt to safe the manipulator system by use of the hardwired control. Note that hardwired control is only a supplement to the first level of hazard control. If the manipulator system cannot be safed by use of the hardwire control, the mission will be assessed to determine if enough time remains to perform an EVA to safe the manipulator system. If hardwired control cannot safe the manipulator system and time does not permit an EVA to safe the manipulator or remove it for stowage, then the RMS will grapple the telerobot using the RMS grapple fixture for jettison. This is the second method for hazard control. The third method of hazard control to provide two fault tolerance for Orbiter safety is EVA operations. Remedial operations could be to remove the manipulator, release the gripper and/or release the actuator brakes. This is to allow stowage of the manipulator, either into its caging devices or by removal and strapping it in the airlock, or otherwise by release into orbit.

Hardwired Control

The FTS system incorporates a backup hardwired control capability in the event of a failure which precludes closed loop computer control of the manipulator system. The main purpose is to minimize the likelihood of having to jettison the system or perform an EVA operation. This has the effect of making the computer system, sensor systems, software, servo systems and most other hardware single fault tolerant, even though the operations would be significantly degraded in performance.

Operational use of the hardwired control is limited to safing of the system after a failure, by stowing the arm to allow a safe Orbiter return. It allows operator control of individual manipulator joints for stowage and for gripper actuation in the event of computer control or motor drive failure. When selected, primary power is removed from all manipulator motor and brake drivers while retaining power to camera controls. Software recognizes the status of the hardwire control, and commands off all motors and brakes, so that return to normal computer operations after hardwired control starts with all motors and brakes powered off.

Hardwire control is limited to very low joint rates and torques in a two fault tolerant manner. Hardwired control is by sequential, joint-by-joint movement, and provided no force accommodation to minimize forces imparted into interfaces. Only a limited set of initiated tasks are likely to be able to be completed. Emergency shutdown detection (ESD) is not operational during hardwired control operation, as the operator can de-power the hardwire drive to stop payload motion, and brakes can also be used to stop motion.

EVA Operations

Several failures of components employ EVA as the third fault tolerant paths to ensure stowage of DTF-1 for safe return of the Orbiter. The manipulator actuators, gripper mechanism, and manipulator caging mechanisms represent major groups of such components.

Failure of a caging mechanism to release the arm for operation would not require EVA for safing the manipulator. EVA would be used as the third path for safing the manipulator if more than one of the four caging mechanism fail to close. In this case, removal of the manipulator at its shoulder interface and either manual release into orbit or stowage in the airlock would be required.

Failure of a manipulator actuator motor drive electrically or mechanically would require EVA as the third fault tolerant path. Mechanical release of the joint actuator brake allows EVA backdrive of the joint into the caging position. If a manipulator joint seizes, then EVA is employed as the third fault tolerant path to remove the manipulator at the shoulder and release into orbit or stowage in the airlock.

Single-Points Failures:

There are several single point failures that remain in the FTS system which may lead to failure of the manipulator to complete a task, or to stow itself for a safe Orbiter return. For the Orbiter this is considered a catastrophic hazard, therefore the requirements for payloads to provide two fault tolerant methods of dealing with these effects.

The FTS single-point failures which lead to an EVA or jettison are few in function, but have commonality within the actuator and gripper. These failures are seized bearings or gears, a short within the motor winding, or a short or open in a brake winding.

Safety Critical Subsystems

The DTF-1 Flight Experiment of FTS has fifteen different safety critical subsystems and equipment groups, as listed.

Structure Subsystem
Control
Data Management and Processing
Vision
Software
End-of-Arm Tooling
Task Panel Elements
Hand Controllers
Thermal Control
Electrical
Power
Sensors
Manipulator
Electromechanical Devices
Aft Deck Workstation

This is only a listing, descriptions of these subsystems will be presented in a future paper.

REFERENCES

Shattuck, P. L., and Lowrie, J. W., Flight Telerobotic Servicer Legacy, SPIE Vol. 1829 Cooperative Intelligent Robotics in Space III, 1992, pg. 60 - 74.

Andary, J. F., Hewitt, D. R., and Hinkal, S. W., The Flight Telerobotic Servicer Tinman Concept: System Design Drivers and Task Analysis, Proceedings of the NASA Conference on Space Telerobotics, Vol. III, January 31, 1989, pg. 447 - 471.

Flight Telerobotic Servicer Development Test Flight (DTF-1) Final Report, MD-15, Martin Marietta Astronautics Group, submitted to NASA Goddard Space Flight Center under contract NAS5-30689, 1991.

Space Station Flight Telerobotic Servicer (FTS) Phase C/D DTF-1 Phase II Safety Compliance Data Package for Payload Design and Flight Operations, 01-PA-32-08, Martin Marietta Astronautics Group, submitted to NASA Goddard Space Flight Center under contract NAS5-30689, August 1, 1991.

Space Station Flight Telerobotic Servicer (FTS) DTF-1 System Critical Design Review, 01-MD-02-CDR-02, Martin Marietta Astronautics Group, submitted to NASA Goddard Space Flight Center under contract NAS5-30689, October 2, 1990.

Space Station Flight Telerobotic Servicer (FTS) Design Criteria Document, 87600000005 Rev. N, Martin Marietta Astronautics Group, submitted to NASA Goddard Space Flight Center under contract NAS5-30689, March 23, 1990.

Space Station Flight Telerobotic Servicer (FTS) DTF-1 Manipulator Subsystem Specification - Final, 01-TR-20-MS-06, Martin Marietta Astronautics Group, submitted to NASA Goddard Space Flight Center under contract NAS5-30689, December 4, 1991.

Space Station Flight Telerobotic Servicer (FTS) DTF-1 Control Stability Analysis and Simulation Report, 01-TR-05-02, Martin Marietta Astronautics Group, submitted to NASA Goddard Space Flight Center under contract NAS5-30689, October 11, 1990.

Space Station Flight Telerobotic Servicer (FTS) DTF-1 Control Stability Analysis and Simulation Report - Preliminary, 01-TR-05-01, Martin Marietta Astronautics Group, submitted to NASA Goddard Space Flight Center under contract NAS5-30689, July 10, 1989.

N94- 34035

EVA-SCRAM OPERATIONS

presented to

Space Technology Interdependency Group (STIG)

at the Seventh Annual

Space Operations, Applications, and Research (SOAR) Symposium

August 3-5, 1993

NASA Johnson Space Center, Houston, Texas

by

David Tamir
Rockwell International Corporation
Space Systems Division
Mail Code ZK32
Kennedy Space Center, FL 32815
Tel. (407) 861-4744

Lee A. Flanigan
Rockwell International Corporation
Rocketdyne Division
950 Explorer Blvd., Suite 3B
Huntsville, AL 35806
Tel. (205) 544-2643

Jack L. Weeks
Rockwell International Corporation
Rocketdyne Division
950 Explorer Blvd., Suite 3B
Huntsville, AL 35806
Tel. (205) 544-2741

Sidney R. McClure
NASA Goddard Space Flight Center
Aerospace Welding Laboratory
Mail Code 752.1
Greenbelt, MD 20771
Tel. (301) 286-2103

Andrew G. Kimbrough
Dimetrics Incorporated
Talley Industries
404 Armour St.
Davidson, NC 28036
Tel. (704) 892-8872

ABSTRACT

This paper wrestles with the on-orbit operational challenges introduced by the proposed Space Construction, Repair, and Maintenance (SCRAM) tool kit for Extra-Vehicular Activity (EVA). SCRAM undertakes a new challenging series of on-orbit tasks in support of the near-term Hubble Space Telescope, Extended Duration Orbiter, Long Duration Orbiter, Space Station Freedom, other orbital platforms, and even the future manned Lunar / Mars missions. These new EVA tasks involve welding, brazing, cutting, coating, heat-treating, and cleaning operations. Anticipated near-term EVA-SCRAM applications include construction of fluid lines and structural members, repair of punctures by orbital debris, refurbishment of surfaces eroded by atomic oxygen, and cleaning of optical, solar panel, and high emissivity radiator surfaces which have been degraded by contaminants. Future EVA-SCRAM applications are also examined, involving mass production tasks automated with robotics and artificial intelligence, for construction of large truss, aerobrake, and reactor shadow shield structures. Realistically achieving EVA-SCRAM is examined by addressing manual, teleoperated, semi-automated, and fully-automated operation modes. The operational challenges posed by EVA-SCRAM tasks are reviewed with respect to capabilities of existing and upcoming EVA systems, such as the Extra-vehicular Mobility Unit, the Shuttle Remote Manipulating System, the Dexterous End Effector, and the Servicing Aid Tool.

INTRODUCTION

Today, we do not have enough on-orbit construction, repair, and maintenance capabilities to effectively support aggressive space programs: such as Hubble Space Telescope (HST), Extended Duration Orbiter (EDO), Long Duration Orbiter (LDO), Space Station Freedom (SSF), other orbital platforms, and manned Lunar / Mars missions. Therefore, it's critical that we expand our on-orbit capabilities and develop new tools to deal with the more demanding tasks that lie closely ahead. The Space Construction Repair and Maintenance (SCRAM) tool-kit will provide us with some of the tools needed to prevail through our space programs, and eventually help us conquer the space frontier (see Figure 1). Employing extra-vehicular activity (EVA) SCRAM tools presents new challenges with on-orbit operations. This paper will focus on EVA-SCRAM's applications and the corresponding on-orbit and even Lunar surface scenarios and performance and safety issues. ***

Figure 5 - Need for Space Construction, Repair, and Maintenance (SCRAM) Tools

EVA-SCRAM DEFINITIONS

This paper will employ the following definitions, some of which are specifically tailored for this paper's discussion. **EVA-SCRAM** encompasses construction, repair, and maintenance which occur outside the pressurized hull of the spacecraft (in-vacuum), whether it's on-orbit or on the Lunar surface. **On-orbit** includes low Earth orbit (LEO), Lunar transfer, Lunar, Mars transfer, and Mars orbits. **In-Space** includes both on-orbit and Lunar-surface operations. **EVA-SCRAM operations** do not have to involve EVA crewmembers, they may be executed by telerobotics alone. **Telerobotics** means that telepresence is being used to operate a robotic slave arm. **Telepresence** (teleoperation) means that sensor feedback (i.e. visual) from the EVA-SCRAM worksite is relayed real-time to a crewmember inside the spacecraft, so he may remotely assist or execute an EVA-SCRAM operation. A **robotic** slave arm may be teleoperated or fully-automated. **Full-automation** means that artificial intelligence is being employed by a machine to execute a set of continuous tasks, requiring no human intervention. **Artificial Intelligence** means that the fully-automated device employs sensors to gather real time feedback on the operation, and accordingly be capable of adapting the operation parameters to dynamic factors. **Semi-automation** means that a task is accomplished by a pre-programmed device, which has been manually or telerobotically set-up and activated. **Manual** means that the EVA crewmember has to use his own body (i.e. hands, senses) to accomplish a task. **Operation modes** include: (1) manual, (2) teleoperation, (3) semi-automation, (4) full-automation. In summary, the following are potential **EVA-SCRAM operation modes**: (1), (1+2), (1+3), (1+2+3), (2), (2+3), or (4).

EVA-SCRAM CAPABILITIES

Since the 1960's, extensive research and development (R&D) efforts have occurred, trying to achieve on-orbit welding capability. Consequently, several thermal processes have been investigated and are

*** Note: Detailed discussion of the need, processes, development, and description of the SCRAM tool-kit is presented in a separate paper at this Symposium, titled "The SCRAM Tool-Kit."

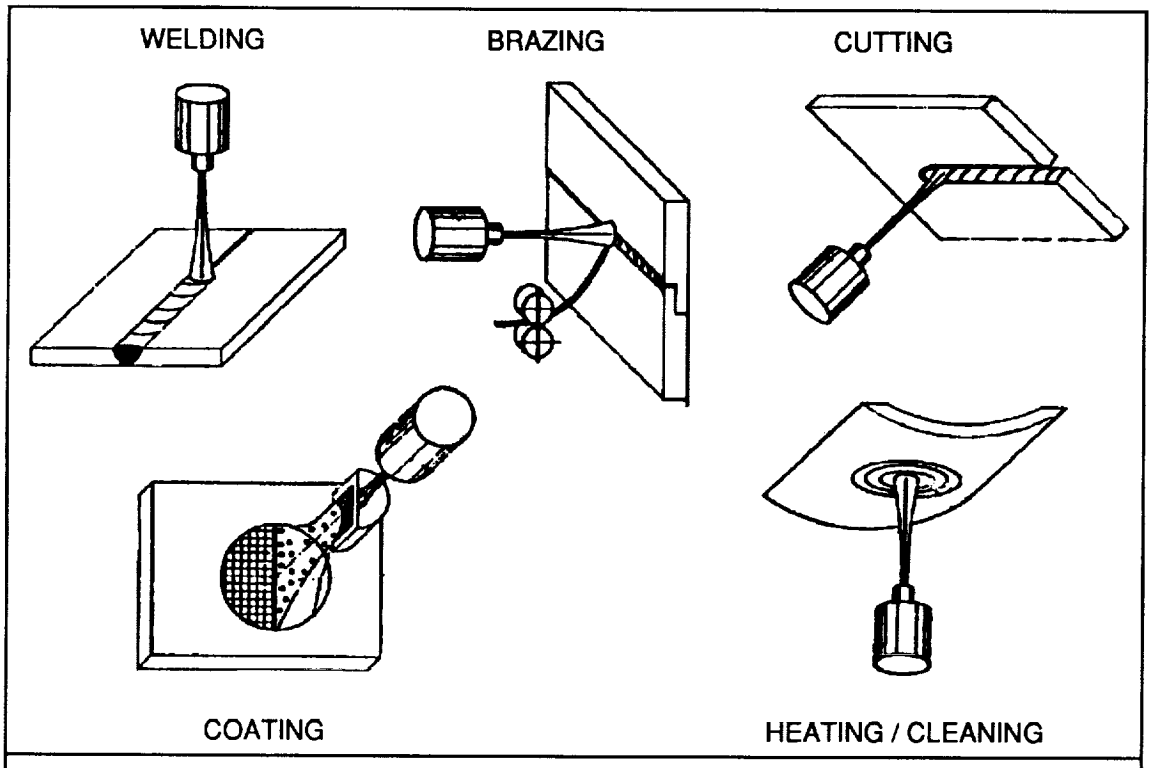


Figure 2 - Space Welding and Thermal Tool (SWATT) Capabilities

still being refined today. These include electron beam, gas tungsten arc, plasma arc, and laser beam. In addition to welding, however, other capabilities have been shown feasible with these thermal processes. Accordingly, this paper employs the term Space Welding and Thermal Tools (SWATT) to collectively identify several multi-function processes (tools) which are capable of welding, brazing, cutting, coating, heating, and even cleaning (see Figure 2). The SCRAM tool-kit has been devised to house SWATT and its complementary quality assurance and control tools, which would perform on-orbit in-situ non-destructive examination (NDE) of the workpiece. SCRAM's NDE tools employ electrical, ultrasonic, x-ray and optical processes (i.e. eddy current, EMAT, radiography, laser refraction). SCRAM would also house work-piece surface preparation tools and set-up assemblies.

EVA-SCRAM APPLICATIONS

EVA-SCRAM's SWATT and NDE capabilities of welding, brazing, cutting, coating, heating, cleaning, and inspection lend themselves to various applications in near-term Shuttle and SSF missions, and in future manned Lunar / Mars missions. The EVA-SCRAM applications set various new challenges for manual, teleoperated, semi-automated, and fully-automated EVA operation modes, both on-orbit and on the Lunar surface.

Shuttle Missions: On-going Shuttle missions carry an EVA tool-kit for in-flight contingencies. This kit is called Provisions Stowage Assembly (PSA) tools, and is stowed in the cargo bay. EVA-SCRAM tools will complement and improve the PSA's existing repair capabilities during contingencies. Longer duration Shuttle missions (EDO / LDO), with on-orbit stays reaching 30 to 90 days, will need to be capable of repairing punctures by orbital debris or damage by fatigue to the crew compartment, Spacelab module, tunnel adapter, external airlock, radiator panels, or vehicle structure (i.e. cargo-bay doors and latching mechanisms). In addition, shuttle servicing missions of LEO platforms and satellites could employ EVA-SCRAM for repair and maintenance of these spacecraft (i.e. cleaning of HST optics). EVA-SCRAM tools could be employed with Shuttle missions via a combination of manual, semi-automated, and telepresence techniques (see Figure 3). Direct teleoperation of EVA-SCRAM tools may also be feasible, should the Shuttle arm, the remote manipulating system (RMS), be improved for more

dexterous operations (i.e. with the Dextrous End Effector now under development). In addition, teleoperation or even full automation of EVA-SCRAM may be achievable using a dedicated robotic slave arm (i.e. the Servicing Aid Tool, also under development).

SSF Missions: SSF will present multiple opportunities for repair, maintenance, and construction over its life-span. EVA-SCRAM tools would become critical for repair of orbital debris- or fatigue- damaged habitation / laboratory modules, radiators, pressurized fluid systems, and structure (see Figure 4). Maintenance of surfaces eroded by atomic oxygen or degraded by contamination, and construction of modifications or expansions to the station structure, habitation / laboratory modules, and power and thermal systems will become a routine well suited for EVA-SCRAM.

Lunar Outpost Missions: The imminent renewal of manned Lunar missions will open a myriad of opportunities for EVA-SCRAM to be heavily employed in construction, repair, and maintenance of structures, habitation / laboratory modules, antennae, solar collector arrays, power plants, fluid lines (plumbing), surface vehicles, descent-ascent vehicles, and other various equipment (see Figure 5-6).

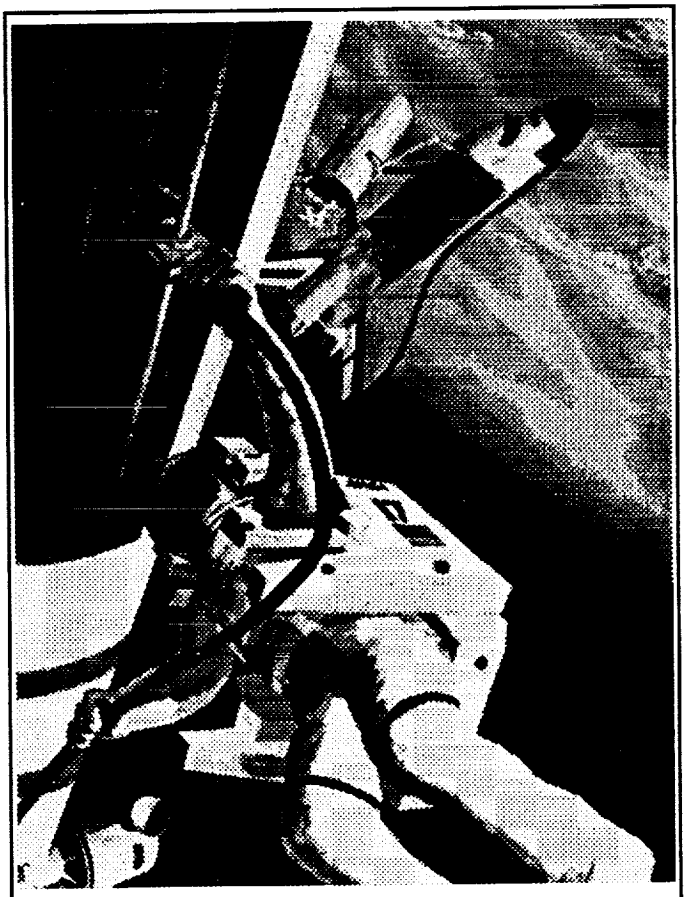


Figure 3 - Shuttle Servicing of Orbital Platforms

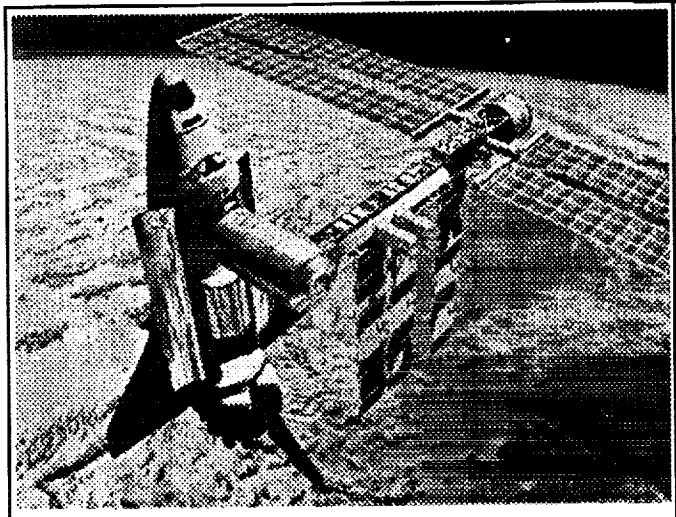


Figure 4 - SSF Construction, Repair, and Maintenance

Figure 5 - Lunar Outpost Construction

Figure 6 - Lunar-Based Antennae Construction

Manned Mission to Mars: The eventual manned missions to Mars will consist of LEO preparation, interplanetary transfer, low Mars orbit, landing and exploration, and return to Earth phases. Over all these phases, manned Mars missions could employ EVA-SCRAM tools on the orbital transfer, descent,

ascent, and surface vehicles. The vehicles' construction, repair, and maintenance tasks suited for EVA-SCRAM will involve structures, habitation / laboratory modules, aerobrakes, antennae, solar collector arrays, radiators, power plants, nuclear shadow shields, fluid lines, and various other equipment (see Figures 7-8).

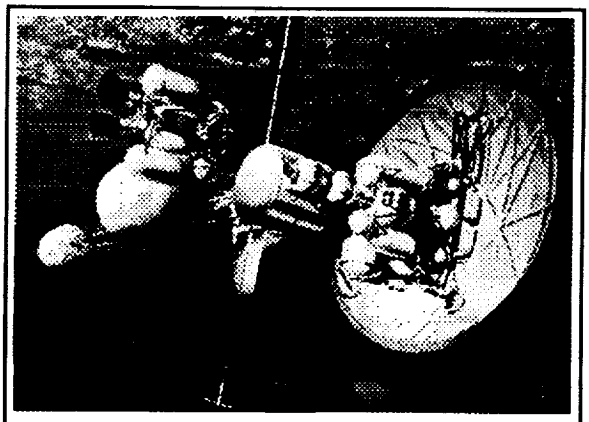


Figure 7 - Mars Orbital Transfer Vehicle's Aerobrake Construction

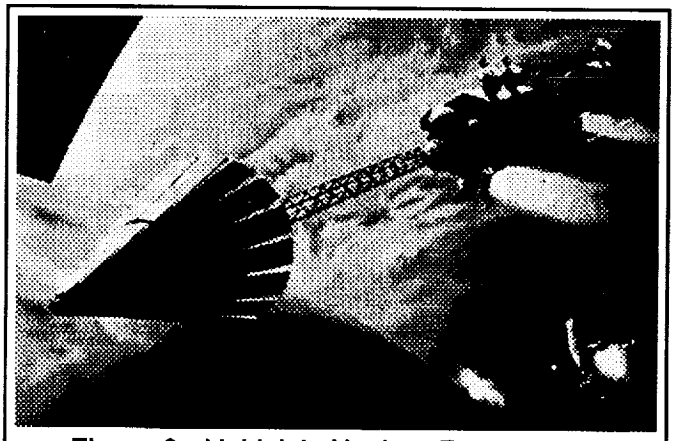


Figure 8 - Vehicle's Nuclear Reactor Shadow Shield Construction

EVA-SCRAM SCENARIOS

Nearest-term EVA-SCRAM operations will include construction of fluid lines, construction of structural members, repair of orbital-debris punctures, refurbishment of surfaces eroded by atomic oxygen, and cleaning of contaminated optics solar panels, and high emissivity radiator surfaces. Additional future EVA-SCRAM operations will include mass production tasks, such as construction of large orbital trusses, aerobrakes, nuclear reactor shadow shields, and Lunar outpost structures. These projected tasks (near-term and future) introduce unique scenarios for both on-orbit and Lunar surface EVA operations. It is anticipated that the near-term EVA-SCRAM operations, which may begin as soon as the late 1990's, will rely heavily on semi-automated devices supported by manual or telerobotic set-up and manipulation (see Figures 3, 9, and 10). Eventually, however, artificial intelligence will enable fully-automated robotic EVA-SCRAM operations. The following eight scenarios will establish typical operational challenges which will have to be mastered.

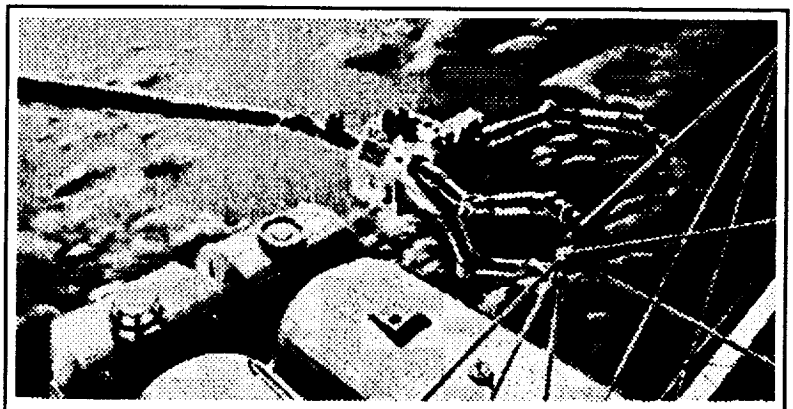


Figure 9 - Robotic SCRAM Operations

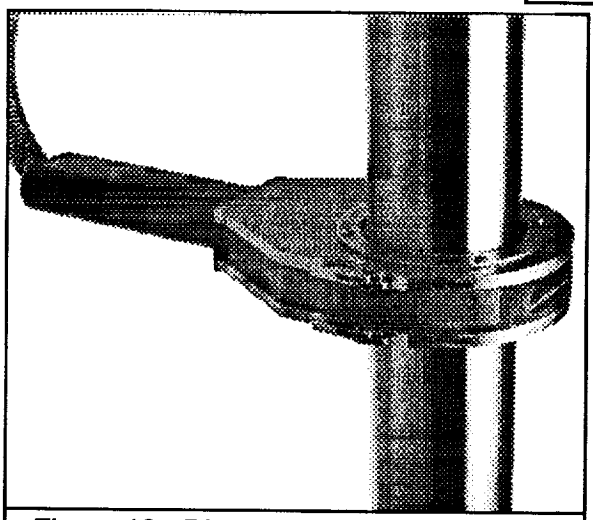


Figure 10 - Dimetric's Commercial GTA Semi-Automated Orbital Welding Device

Scenario-I: (Fluid Line Construction) - This scenario involves assembly of tubular lines or ducting which may be used for thermal control, propulsion, venting, life support, and laboratory supplies. The work-piece materials are likely to involve aluminum, stainless steel, titanium, and Inconel alloys. Due to standard geometries exhibited by tubular lines (varying primarily in diameter and wall thickness), most EVA-SCRAM operations (i.e. cutting, welding, NDE) in this scenario

would be executed using semi-automated orbital devices which may be set-up manually or telerobotically (see Figures 3, 9, and 10).

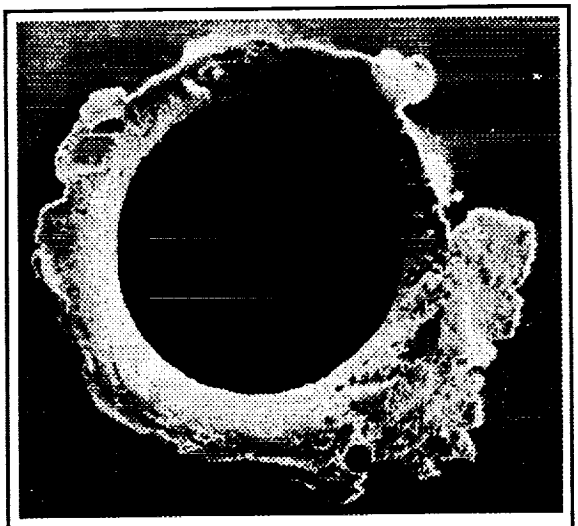


Figure 11 - 0.5 mm Diameter Puncture Found in a Spacecraft Louver

Scenario-II: (Structural Member Construction) - This scenario involves assembly of structural members which may be in the form of brackets, struts, beams, small truss, tubular extrusions, or plates. These members may be used for mounting equipment to the outside of the spacecraft, for routing and housing electrical lines (i.e. electrical conduit), or for shielding spacecraft systems. The work-piece materials are likely to involve aluminum, stainless steel, titanium, and Inconel alloys, and even composites. This scenario may involve both standard and non-standard geometries at the structural joints to be welded or cut. For standard geometries (i.e. involving tubular extrusions), semi-automated devices can be employed as described in Scenario-I. For non-standard geometries, automated joint seam-tracking may have to be employed using teleoperation, robotics, and artificial intelligence. Rockwell's Rocketdyne division has developed artificial intelligence capabilities integrated with robotics for complex welding tasks of Space Shuttle Main Engine components.

Scenario-III: (Orbital-Debris Puncture Repair) - This scenario involves repair of spacecraft surfaces and components which have been punctured by collisions with orbital debris. Punctures are most likely to be of small diameter, probably between 0.5 to 5 mm (see Figure 11); however, larger holes are possible. The workpiece materials are likely to involve aluminum, stainless steel, titanium, and Inconel alloys, and even composites. Punctures to pressurized systems (i.e. crew-laboratory module, radiator panel, fluid line, fuel tank) may require depressurization of the system prior to repair, due to interfering leakage. This type of repair scenario may employ SCRAM's surface preparation (i.e. cleaning, cutting) and welding and coating capabilities. Standardized circular patches may be employed with a semi-automated orbital device which would perform the repair operation after manual or telerobotic set-up.

Scenario-IV: (Atomic Oxygen Erosion Refurbishment) - This scenario involves re-coating spacecraft surfaces which have been eroded by atomic oxygen bombardment. Such surfaces involve thermal control radiators, telescope mirrors, electric conductors, and transmission or receiving antennae. The workpiece materials are likely to involve aluminum, stainless steel, titanium, Inconel, quartz, and various other materials. This scenario would most likely involve re-coating significant areas, lending itself to an automated operation. Very high dexterity motion (i.e. as with welding) would probably not be required. An automated robotic tool (i.e. like those used for spray-painting automobiles on a terrestrial assembly line) may be applied effectively with a manual or telerobotic set-up (i.e. using RMS).

Scenario-V: (Surface Cleaning) - This scenario involves cleaning spacecraft optical, solar collector, and thermal control surfaces, which are permanently exposed to the extra-vehicular space environment. Performance of windows, mirrors, lenses, high emissivity radiator surfaces, and solar panel surfaces are gradually degraded by polymerized and cross-polymerized organic contamination (hydrocarbons and siloxanes), generated by exposure to the vacuum and ultraviolet radiation environment. These contaminants are also generated by spacecraft outgassing products, fuel, and propulsion by-products. This scenario would most likely involve cleaning significant areas, lending itself to an automated operation (similar to scenario-IV). Very high dexterity motion (i.e. as with welding) would probably not be required. An automated robotic tool (i.e. like those used for spray-painting automobiles on a terrestrial assembly line) may be applied effectively with a manual or telerobotic set-up.

Scenario-VI: (Large Orbital Truss Construction) - This scenario involves assembly of large truss structures by mass production welding and NDE of truss member joints (see Figures 1 and 12). The work-piece materials are likely to involve aluminum, stainless steel, titanium, and Inconel alloys, and

even composites. For this scenario's repetitive and tedious mass production operations, automation is preferred. Truss members joints can be made suitable for automated EVA-SCRAM operations. For example, tubular truss members can be effectively welded and inspected with an orbital EVA-SCRAM device, like the one proposed for scenario-I (see Figure 10). To achieve full automation, however, the orbital device should be manipulated (set-up on the workpiece) by an autonomous robotic system with built-in artificial intelligence (see Figure 9). Less efficient truss construction can be achieved using telerobotic manipulation of the EVA-SCRAM device (i.e. using the RMS), or even manual manipulation. However, the actual joint seam tracking (i.e. for welding and inspection) will have to be accomplished using automation.

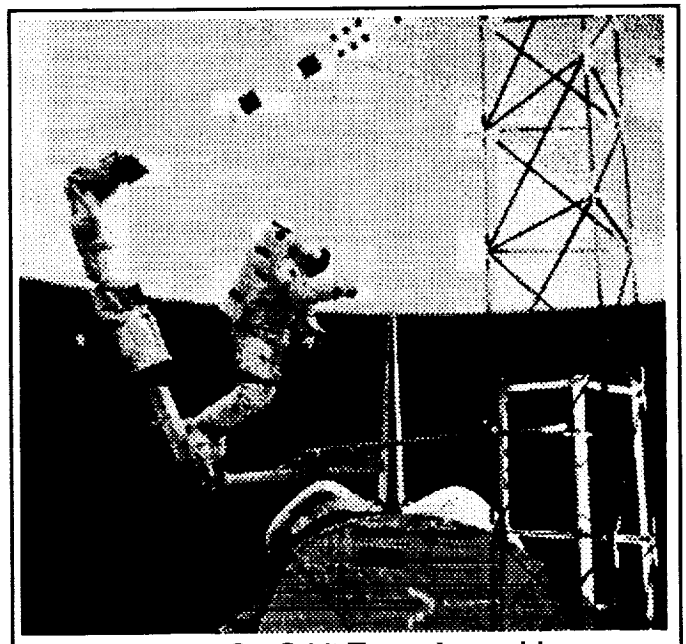


Figure 12 - On-Orbit Truss Assembly

Scenario-VII: (Aerobrake and Shadow Shield Construction) - This scenario involves assembly of large plated-structures by mass production welding and NDE of plate member joints (see Figures 7 and 8). The work-piece materials are likely to involve aluminum, stainless steel, titanium, and Inconel alloys, and even composites. For this scenario's repetitive and tedious mass production operations, automation is preferred. Plate member joints can be made suitable for automated EVA-SCRAM devices. To achieve full automation, however, the EVA-SCRAM device should be manipulated (set-up on the workpiece) by an autonomous robotic system with built-in artificial intelligence (see Figure 9). Less efficient construction can be achieved using telerobotic manipulation of the SCRAM device (i.e. using the RMS), or even manual manipulation. However, the actual joint seam tracking (i.e. for welding and inspection) will have to be accomplished using automation.

Scenario-VIII: (Lunar Outpost Construction) - This scenario involves a very important element which has been absent from the previous seven scenarios - gravity. The Moon's gravitational force, even though about one sixth of Earth's, will greatly facilitate EVA-SCRAM operations. The EVA-SCRAM operator no longer has to be tethered to the tools and attached, in one form or another, to a spacecraft. In addition, relative motion and position between the EVA-SCRAM operator and the workpiece will be simpler to control, since the Moon's gravity will bound both. Lunar EVA-SCRAM operations will involve various applications as described earlier, including construction, repair, and maintenance of structures, habitation / laboratory modules, antennae, solar collector arrays, power plants, fluid lines (plumbing), surface vehicles, descent-ascent vehicles, and other various equipment (see Figure 5-6). Work-piece materials are likely to involve aluminum, stainless steel, titanium, and inconel alloys, and even indigenous lunar produced metal alloys. Since the Lunar surface is still a vacuum-radiation hostile environment for EVA manned operations, execution of EVA-SCRAM tasks would emphasize teleoperation and automation over manual operations. However, some manual EVA support will always be required.

SCRAM PERFORMANCE AND SAFETY ISSUES

Designing an EVA-SCRAM system, requires that we consider the performance and safety of both the operator (i.e. astronaut, robotic device) and the mission (i.e. remaining crew, spacecraft, payloads).

EVA Crewmember Manual Performance Issues: Three major factors that may degrade EVA crewmember performance are extravehicular mobility unit (EMU) encumbrances, insufficient working volume, and inadequate restraints and mobility aids (see Figure 13). The EMU (space suit) is an independent anthropomorphic system that provides crewmembers with environmental protection, life

support, mobility, communications, and visibility while performing various EVA's. The EMU incorporates a specially designed garment which can withstand high temperature, pressure, radiation, and physical wear. Consequently, the EMU limits the astronaut's manual dexterity and body movement. The EVA crewmember reaches fatigue levels much quicker than an operator (i.e. welder) on Earth; because delicate and precise hand movement require significant mental concentration and physical effort with the impeding pressurized space garment. Manual tasks such as the manual removal or replacement of threaded fasteners, continuous force-torque application, and extended gripping functions need to be minimized by the design of an EVA-SCRAM system (see Figure 14) [ref-1]. In addition, the EMU helmet impedes coordination and severely restricts visual examination of small-tight operations (i.e. welding). Because manual welding is a process which requires precision, coordination, and the ability to control several factors simultaneously (i.e. welding travel speed, welding arc gap, welding current output, and even welding filler wire feeding), the astronaut would have to be extensively prepared and trained for each single task. But, even then manual EVA-SCRAM welding operation may have low (20%) success rate, as can be expected of a comparable challenging terrestrial manual welding task. On Earth we accommodate the low success rate by simply cutting out the unacceptable weld and trying again; however, this would not be practical for space-based operations. Consequently, EVA-SCRAM operations must rely on the use of semi-automated devices, which would require relatively simple manual or telerobotic set-up. The EVA-SCRAM processes (welding, brazing, cutting, coating, cleaning, and NDE - see Figure 2) should be automatically executed with computer adaptive control, similar to existing terrestrial orbital welding systems (see Figure 15). These automated systems provide a 98% success rate. Therefore, manual EVA-SCRAM process execution (i.e. manual welding) should be reserved only for contingencies, where an automated system has failed or cannot be applied.

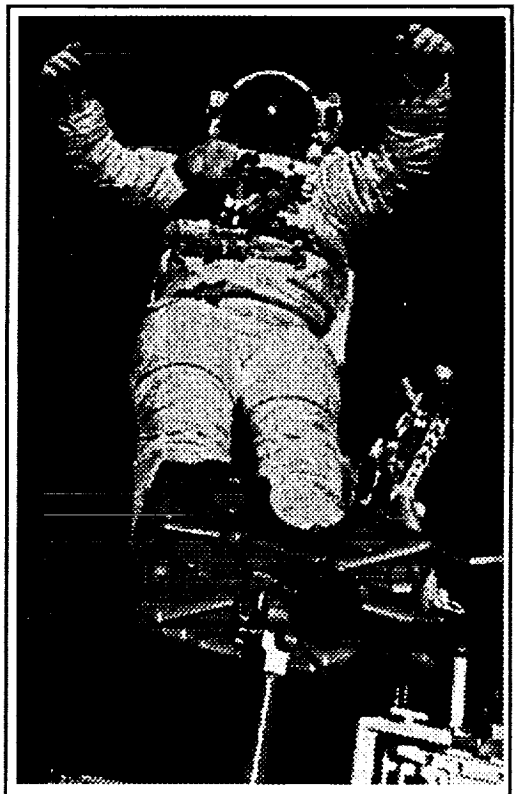


Figure 13 - Suited EVA Crewmember Employing a Portable Foot Restraint

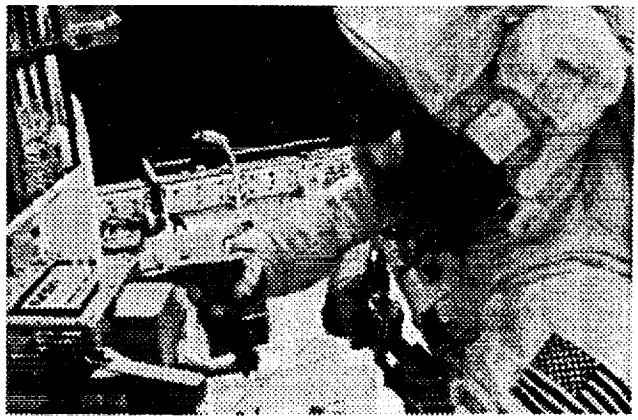


Figure 14 - Limited Manual EVA Dexterity

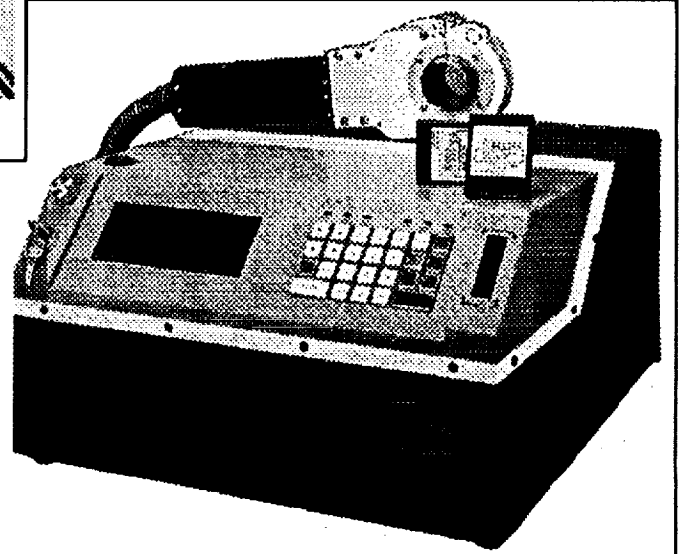


Figure 15 - Dimetric's State-of-the-Art Automated Programmable Welding System

Other Performance Issues: EVA-SCRAM operations will occur in both light and shadow (at 45 minute intervals in LEO) with consequent thermal gradients and sun-light reflections off of the workpiece. These dynamic factors will challenge operation of the EVA-SCRAM pro-

cesses, requiring real-time electronic / optical sensor feedback and adaptive computer control in order to effectively perform routine SCRAM tasks. Robotic slave arms, such as the Shuttle's RMS with the Dextrous End Effector (DEE) and the Servicing Aid Tool (SAT), are incapable of the precision movement degree required for directly executing SCRAM's welding, brazing, cutting, and perhaps NDE processes. The robotic slave arms, however, should be capable of supporting telerobotic set-up and activation of a semi-automated EVA-SCRAM device (i.e. orbital welding head - see Figures 10 and 15).

Safety Issues: Safety of the spacecraft crewmembers and mission are of prime importance and, therefore, will govern the design of the EVA-SCRAM tool-kit and its operation modes. EVA-SCRAM's various tools and processes exhibit the following safety hazards: temperature extremes, electrical shock, contamination, and radiation. EVA-SCRAM's SWATT processes (electron beam, gas tungsten arc, plasma arc, and laser beam) are thermal tools which generate hot temperature extremes (i.e. a molten weld puddle). SWATT processes employ high currents or high voltages (depending on the process). Some of EVA-SCRAM's operations (i.e. welding, coating, cutting) produce varying levels of metal vapor which is redeposited on near-by surfaces. Lastly but not least, EVA-SCRAM's SWATT and NDE processes produce various levels and combinations of radiation (depending on the process), including: ultraviolet, infrared, extreme bright light, laser light, x-ray, accelerated electrons, and electro-magnetic interference. In summary, EVA-SCRAM is faced with various challenges of providing acceptable hazard inhibits and controls. SCRAM's terrestrial counter-part processes (i.e. welding) have been employed for years successfully, meeting safety requirements via various conservative safety measures and techniques. These terrestrial safety techniques and others can be modified to solve all of the safety issues associated with EVA-SCRAM.

Figure 16 - Former Soviet Union EVA Manual Welding Experiment

CONCLUSION

EVA-SCRAM is a leap into a new realm of on-orbit and even Lunar surface operations. By taking-on the challenge of EVA SCRAM operations, using the acceptable NASA safety approach, we will develop new critically needed tools for our upcoming space programs. The success of HST, EDO, LDO, SSF, and Lunar / Mars manned missions depends on the availability of EVA-SCRAM capabilities. On-orbit manual EVA-SCRAM experiments have already been initiated by the former Soviet Union (see Figure 16). EVA-SCRAM experiments should be continued with emphasis on productive operation modes, including: teleoperation, semi-automation, and full-automation. The EVA-SCRAM manual operation mode should be reserved for semi-automated device set-up and contingencies. EVA-SCRAM operations can be hazardous, especially if an EVA crewmember is present at the worksite. But, these hazards are containable. The nearest-term EVA-SCRAM applications which should be pursued, include: construction of fluid lines and structural members, repair of punctures by orbital debris, refurbishment of surfaces eroded by atomic oxygen, and cleaning of optical, solar panel, and high emissivity radiator surfaces which have been degraded by contaminants.

REFERENCES

[1] NASA NSTS 07700, Volume XIV, Appendix 7, System Description and Design Data - EVA, 1988
[2] Watson, Rockwell Rocketdyne, "Extra-Vehicular Activity Welding Experiment," contract NAS8-37753, 1989
[3] American Welding Society, Paton Welding Institute, "Welding in Space ...," 1991
[4] Kasayev, Ostrovski, Lapchinsky, Commonwealth of Independent States, "Space Welding," 1992
[5] Tamir, Rockwell Space Systems, "A Path to In-Space Welding ... Using Space Shuttle Small Payloads," 1992
[6] Tamir, Florida Institute of Technology, "In-Space EVA Welding," 1992

N94-34036

HARDWARE INTERFACE FOR ISOLATION OF VIBRATIONS IN FLEXIBLE MANIPULATORS—DEVELOPMENT AND APPLICATIONS*

Davoud Manouchehri
Thomas Lindsay, Ph.D.
Rockwell International, Space Systems Division
12214 Lakewood Blvd.
Downey, CA 90241

David Ghosh, Ph.D.
NASA Langley Research Center
Hampton, VA 23665-5225

ABSTRACT

NASA's Langley Research Center (LaRC) is addressing the problem of isolating the vibrations of the Shuttle remote manipulator system (RMS) from its end-effector and/or payload by modeling an RMS flat-floor simulator with a dynamic payload. Analysis of the model can lead to control techniques that will improve the speed, accuracy, and safety of the RMS in capturing satellites and eventually facilitate berthing with the space station.

Rockwell International Corporation, also involved in vibration isolation, has developed a hardware interface unit to isolate the end-effector from the vibrations of an arm on a Shuttle robotic tile processing system (RTPS). To apply the RTPS isolation techniques to long-reach arms like the RMS, engineers have modeled the dynamics of the hardware interface unit with simulation software.

By integrating the Rockwell interface model with the NASA LaRC RMS simulator model, investigators can study the use of a hardware interface to isolate dynamic payloads from the RMS. The interface unit uses both active and passive compliance and damping for vibration isolation. Thus equipped, the RMS could be used as a telemanipulator with control characteristics for capture and berthing operations. The hardware interface also has applications in industry.

INTRODUCTION

NASA's Langley Research Center and Marshall Space Flight Center (MSFC) have joined forces to study berthing operations between the Shuttle remote manipulator system and Space Station *Freedom* (SSF) by constructing the Coupled, Multibody Spacecraft Control Research Facility at MSFC (Reference 1). This flat-floor test bed is composed of a two-link, three-joint manipulator arm that simulates the RMS and a large, controlled mass that simulates the SSF. The SSF model is equipped with air jets and a torque wheel to simulate the reaction jets and control moment gyro (CMG). Experimental runs on the test bed determined system parameters and natural frequencies. From these parameters, a software model of the flat-floor test bed, generated by $Matrix_X$ software, was validated against the actual performance of the system.

One of the important problems to be solved with this test bed is the isolation of vibrations between the SSF and RMS during berthing operations. The RMS has long, flexible links that are susceptible to unwanted

*This work was partially funded by NASA Langley Research Center Contract NAS1-19243 to Rockwell International's Space Systems Division and monitored by Dr. Raymond Montgomery of LaRC Spacecraft Controls Branch.

vibrations. Shuttle, RMS, and SSF motions, as well as force impacts, can induce vibrations. A system that decouples the dynamics between the RMS and SSF can increase the speed, accuracy, and safety of berthing operations.

Rockwell International's Space Systems Division has developed a mechanical interface unit to decouple the dynamics between two coupled systems (Reference 2). Kennedy Space Center (KSC) is developing a robotic device for Shuttle reprocessing operations. After each flight, each tile on the underside of the Shuttle orbiter must be injected with a hazardous fluid that prevents the tiles from absorbing water. Approximately 17,000 of the tiles can be processed by a mobile robotic vehicle that locates each tile, moves into contact, and injects the tile with the rewaterproofing compound. The elevation arm of the robotic tile processing system lifts the end-effector and brings it near the underside of the Shuttle. The Rockwell interface unit then positions a rewaterproofing nozzle in contact with the Shuttle tiles. As the rewaterproofing compound is injected into each tile, the interface unit must maintain the proper force between the nozzle and the tile, regardless of any vibrations the arm may impart to the interface unit. The interface unit uses active and passive compliance and damping to decouple the vibrations of the arm from the nozzle and the Shuttle tile. A $Matrix_X$ software model of the interface unit was developed to test control algorithms and validate interface unit operations under various loading conditions.

As a feasibility study for using the Rockwell interface unit in the LaRC/MSFC flat-floor test bed, Rockwell and LaRC integrated the $Matrix_X$ software simulators of the test-bed facility and the interface unit. Simulations to date show very promising results, and plans are under way to integrate the hardware unit into the test bed. This paper describes the interface unit hardware, the combined Rockwell/LaRC simulator, and results of the simulations. Future plans and applications are also addressed.

THE ROCKWELL INTERFACE UNIT

The Rockwell interface unit combines one degree of linear actuation (z-direction) with three degrees of passive compliance (linear in the z-direction and rotational about the x- and y-axes). A direct-drive stepper motor with microstepping capability rotates a drive link and connector link that impart motion to the upper platform of the unit. A six-bar linkage is designed to constrain the upper platform to straight-line motion. Springs mounted on the upper platform provide the compliance and a small degree of damping. Mounted on top of the springs, a force/torque sensor sends feedback to the interface unit controller. The payload is mounted to the force/torque sensor. For the RTPS, this payload is the rewaterproofing nozzle. Figure 1 shows the interface unit developed for the RTPS. The total extension length for actuation is 2 inches (49.0mm), and the total travel for the springs is 0.192 inch (4.88mm), allowing for a maximum rotation of 8.40 degrees (0.147 radian). These parameters, as well as the spring constants, are the main variables to be optimized for interfacing with the LaRC/MSFC test bed.

The interface unit developed for the RTPS has been tested under a variety of loading conditions, including vibrations of the interface base while the nozzle is in contact with a solid surface (simulating the Shuttle tile). The interface unit performs quite well and maintains a seal between the nozzle and tile under all operating conditions. However, because the control objectives for the rewaterproofing task differ from the current LaRC/MSFC test-bed operation objectives, a new controller was developed, which is described later.

INTEGRATION OF SYSTEM MODELS

The LaRC/MSFC flat-floor test bed was modeled by LaRC with $Matrix_X$ software. Similarly, Rockwell modeled its interface unit with $Matrix_X$ software. These two software models were integrated, and the combined models are used to simulate the integrated system. Figure 2 illustrates the location of the interface unit in the combined system.

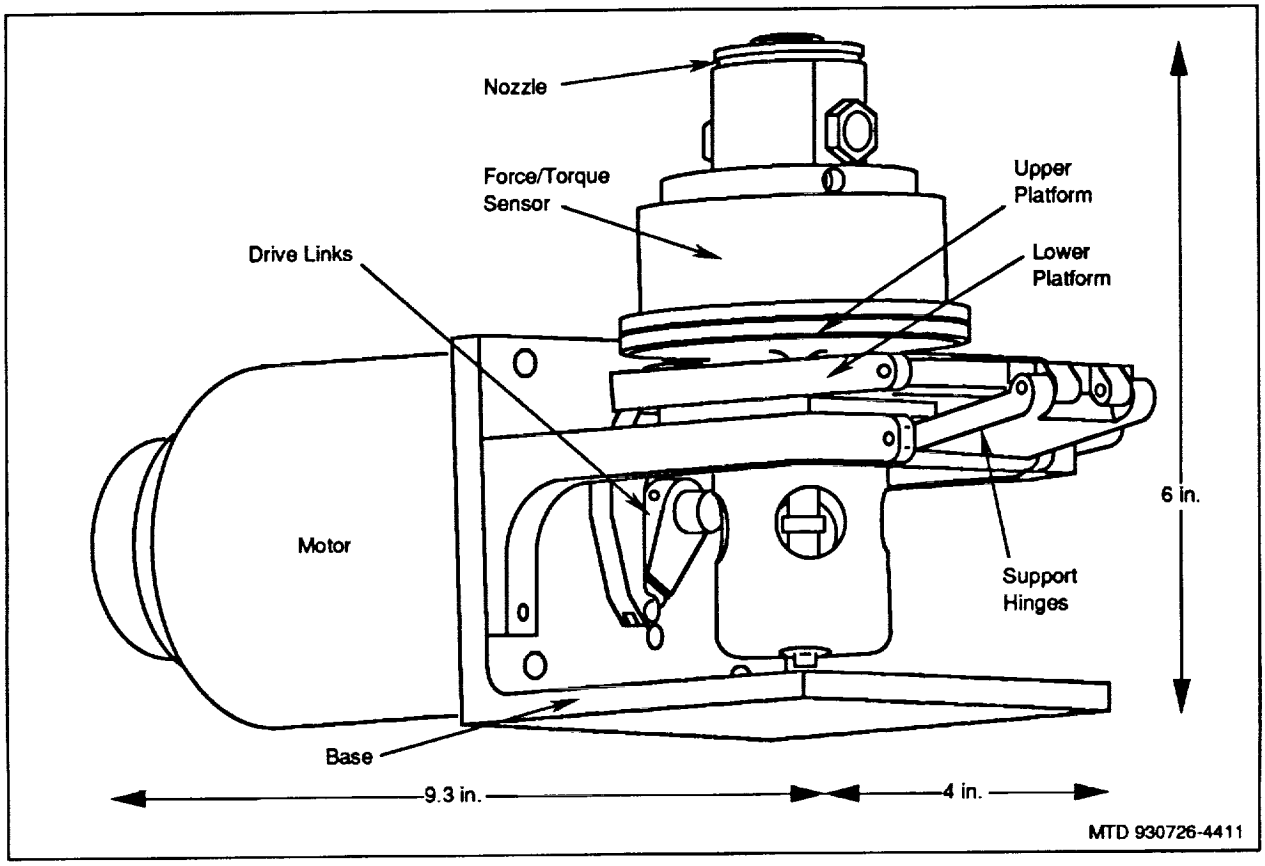


Figure 1—The Rockwell Interface Unit

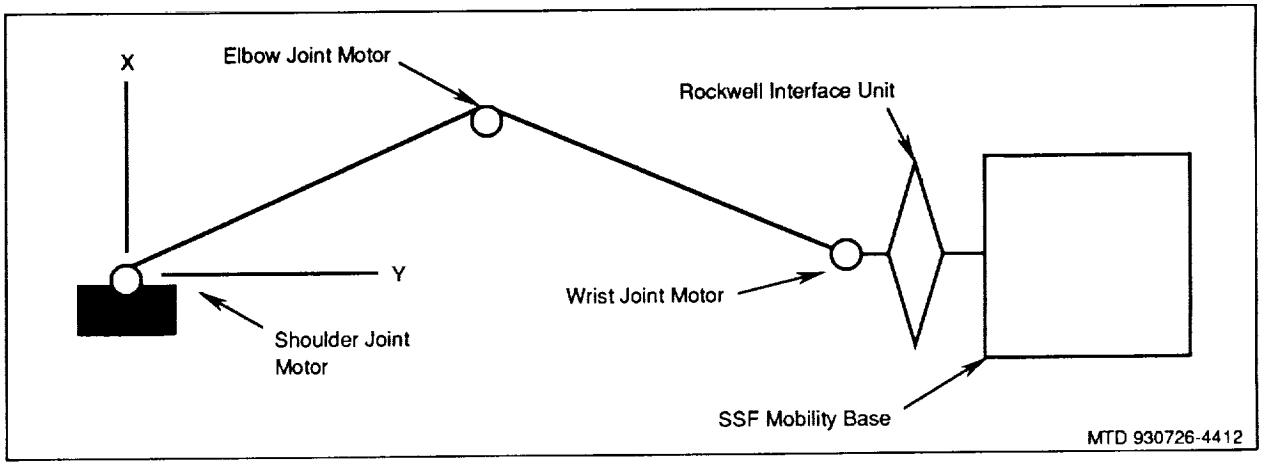


Figure 2—Combined System Layout

The LaRC model is composed of a two-link manipulator with three joints (shoulder, elbow, and wrist), including full motor models, gearing, etc., and a controlled payload mass. To increase computation speed, the dynamics of the payload are calculated along with the dynamics of the manipulator links in a software code block. However, to incorporate the Rockwell interface model, the payload mass was split from the arm dynamics. The interface unit was then installed between the arm and payload.

The Rockwell model began as a stand-alone simulator, and the interface unit was modeled in all three dimensions. Data lines to and from the unit were created, and the model was reduced to two dimensions for faster performance in the two-dimensional flat-floor simulator. The interface unit is subjected to a rotational torque from the manipulator wrist motor, as well as the x- and y-motion of the manipulator end-point. The interface unit applies a rotational torque to the payload and to the length of extension. The payload model returns the rotational position and velocity to the interface, and the interface then returns its rotational position and velocity to the wrist motor. The loading force is also returned from the interface unit to the manipulator arm. Internally, the interface controller responds to the force sensed by the force/torque sensor to control the length of extension.

CONTROL STRUCTURE

The interface unit controller is the most important subject in the ongoing project. One major issue to be resolved is the nature of the control objectives. The first iteration of the interface controller is an attempt to meet two objectives: to minimize the force sensed at the interface and to maintain the position of the interface near its center position. For this controller, only direct readings of the force at the interface and the extension of the interface are needed.

$$\begin{aligned} vel_c &= -F_{f/t} * pgain_f + pgain_p * ((z_{mid}/(z_{max} - z))^2 - 1.0) \quad \text{if } z > z_{mid} \\ &= -F_{f/t} * pgain_f + pgain_p * ((z_{mid}/(z - z_{min}))^2 - 1.0) \quad \text{if } z \leq z_{mid} \end{aligned} \tag{1}$$

$$pgain_f = gain * (z_{mid} - |z - z_{mid}|)/z_{mid} \tag{2}$$

where

vel_c = commanded rotational velocity to interface unit motor

$F_{f/t}$ = z-force from force/torque sensor

z = current extension of interface unit

$pgain_f$ = proportional gain for force feedback

$pgain_p$ = proportional gain for position feedback

z_{mid} = middle position for actuator

z_{max} = maximum extension for actuator

z_{min} = minimum extension for actuator

Toward the extremes of the actuator motion, the effect of a load force diminishes, while the command to restore the central position increases. A steady load force causes the unit to maintain an off-center equilibrium point.

This control algorithm meets two control objectives: attempt to zero out load forces while attempting to maintain a center equilibrium position. These objectives could provide good performance characteristics for many operations. However, meeting other control objectives may become important. For example,

if the manipulator arm performance degrades only for certain load-force frequencies, it may be beneficial to scan sensor inputs for these frequencies and attempt to isolate or damp only these problem frequencies. As another example, monitoring the sensor inputs to determine the load impedance would allow the controller to adapt its gains to either match or mismatch the impedance, depending on the goals of the operation. As a final example, a capture operation may require the interface unit to first comply completely with load forces but then slowly attempt to damp out any vibrations while stiffening the actuator until it finally becomes locked. The most useful interface unit would be completely self-contained, requiring no operator input. However, for different operations it may be necessary to switch manually between control modes.

SYSTEM SIMULATION

The results presented here are from the combined system using the first-iteration controller, as described above. The combined system is compared with the original LaRC simulator model without an interface unit. The results are extremely promising.

An example run is shown in Figure 3. In it, the manipulator arm is fully extended in the x-direction (all joint angles initially zero). The joint 3 motor is commanded with a step torque of 100 N•m for 2 seconds, a step torque of -100 N•m for 2 seconds, and then a zero input for 4 seconds. Figure 4 compares the resultant joint angles for the system with and without the interface unit. Joints 1 and 2 are free to rotate, and the results indicate that these joints are much less affected by the motion of the payload with the

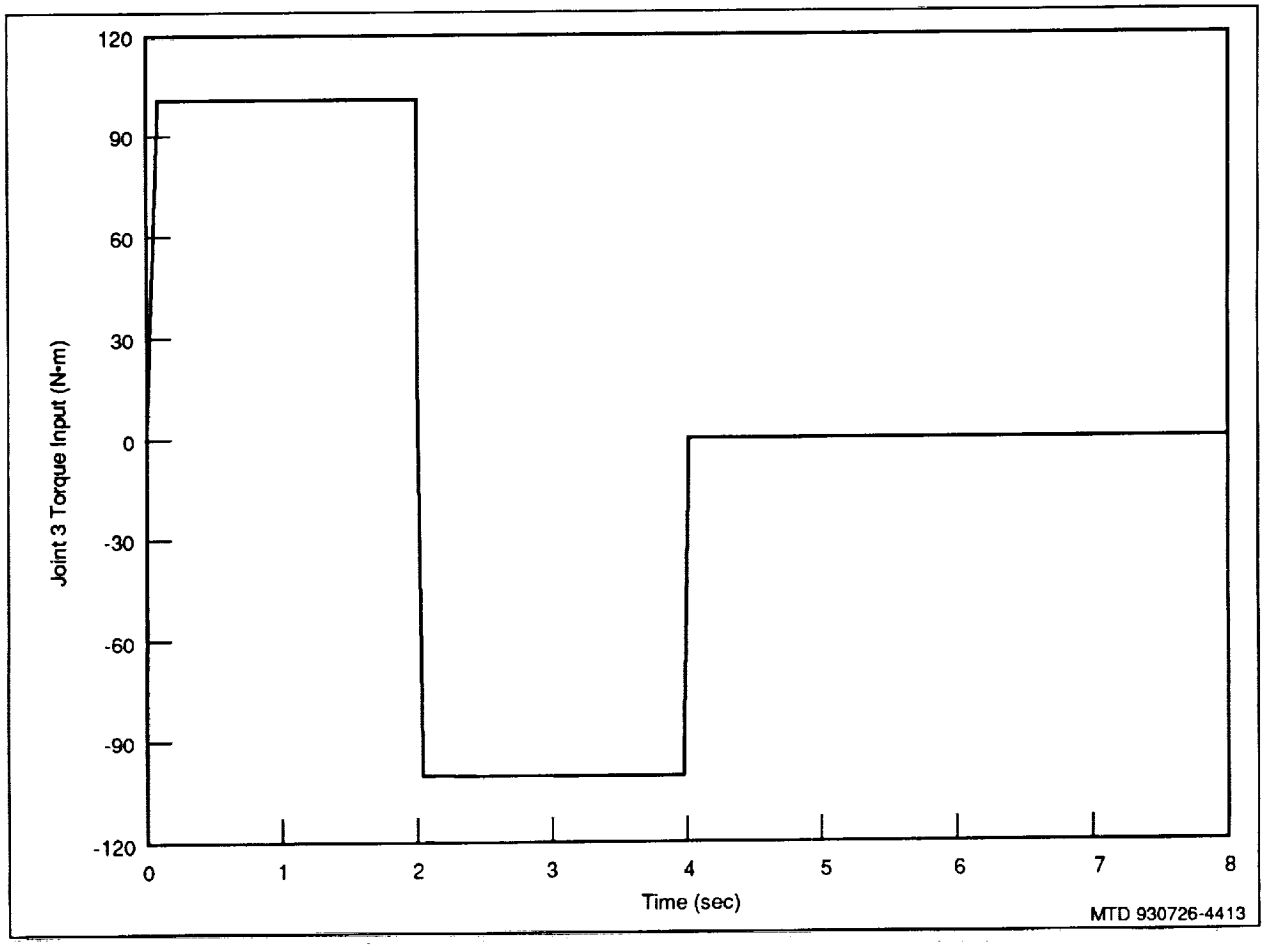


Figure 3—System Input

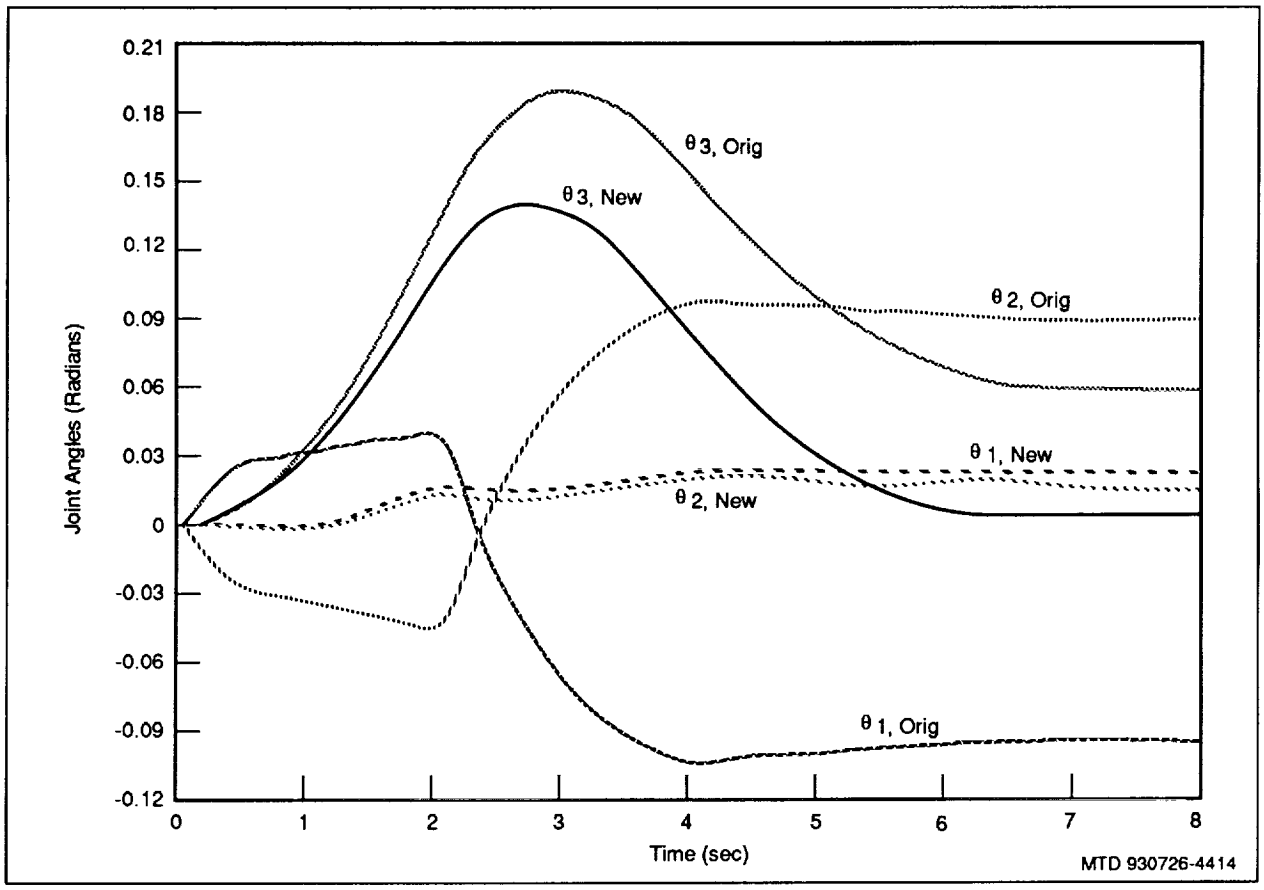


Figure 4—System Response Comparison

interface unit than without. Joint 3 is also less affected by the inertial movement of the payload with the interface unit.

CURRENT AND FUTURE WORK

The software model is currently being updated through the use of a combined dynamic software code structure. This will make the combined model more dedicated to the current task but should increase the computational speed of the simulator and the sensitivity of the simulator to numerical integration instabilities.

The interface unit is being redesigned for integration into the LaRC/MSFC test bed. New control structures are being implemented in the simulation model and then tested with the existing interface unit. An integrated test will soon be performed in the LaRC/MSFC test bed. The results will be used to validate the software simulation and to iterate the interface design for a unit to be permanently integrated into the test bed. Positive results from the test bed unit will prompt the design of a prototype unit for testing on Shuttle missions.

APPLICATIONS

The utility of the interface unit on the Shuttle RMS is evident from the above discussions. However, similar vibration isolation/control and dynamic system decoupling are also needed for other applications of long manipulator arms, including Department of Energy waste cleanup as well as industrial uses. The device

may also prove beneficial for reducing vibrations and impact forces in devices with less accurate control, such as cranes and winches. The implementation of a passive or near-passive interface device for these applications will improve system performance without intelligent human interaction or advanced system control techniques. Further space applications of the device include isolation of antennas and solar panels from a satellite and isolation of payloads from Shuttle vibrations during ascent flight.

REFERENCES

1. Montgomery, Raymond, Tobbe, Patrick, Weathers, John, Ghosh, Dave, and Garrison, James, "A Testbed for Research on Manipulator-Coupled Active Spacecraft," to appear in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Monterey, California, August 9-11, 1993.

2. Manouchehri, Davoud, Hansen, Joseph, Wu, Cheng, Yamamoto, Brian, and Graham, Todd, "Robotic End-Effector for Rewaterproofing Shuttle Tiles," *Proceedings of the SPIE OE/Technology '92 Conference*, Boston, Massachusetts, November 15-20, 1992.

Session R7: ROBOTIC OPERATIONS: SPACE TERRESTRIAL

Session Chair: Mr. Charles Shoemaker

N94-34037

Ground Vehicle Control at NIST: from Teleoperation to Autonomy

Karl N. Murphy, Maris Juberts, Steven A. Legowik, Marilyn Nashman,
Henry Schneiderman, Harry A. Scott, and Sandor Szabo

Robot Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899

ABSTRACT

NIST is applying their Real-time Control System (RCS) methodology for control of ground vehicles for both the U.S. Army Research Lab, as part of the DOD's Unmanned Ground Vehicles program, and for the Department of Transportation's Intelligent Vehicle / Highway Systems (IVHS) program. The actuated vehicle, a military HMMWV, has motors for steering, brake, throttle, etc. and sensors for the dashboard gauges. For military operations, the vehicle has two modes of operation: a teleoperation mode - where an operator remotely controls the vehicle over an RF communications network; and a semi-autonomous mode called retro-traverse - where the control system uses an inertial navigation system to steer the vehicle along a prerecorded path. For the IVHS work, intelligent vision processing elements replace the human teleoperator to achieve autonomous, visually guided road following.

1. INTRODUCTION

NIST's involvement in unmanned ground vehicles started in 1986 with the U.S. Army Research Lab's (ARL, formerly LABCOM) techbase program. This program became part of the Defense Department's Robotics Testbed program resulting in Demo I. NIST's responsibility included implementing a mobility controller and developing an architecture for unmanned ground vehicles (UGV) which would support integration and evaluation of various component technologies. [1,2,3]

In a typical scenario, military personnel remotely operate several Robotic Combat Vehicles (RCVs) from an Operator Control Unit (OCU). Each vehicle contains: actuators on the steering, brake, throttle, transmission, transfer case, and parking brake; an inertial navigation system; a mission package which performs target detection, tracking, and laser designation; and data and video communication links. The OCU contains controls and displays for route planning, driving, operation of the mission package, and control of the communication links.

A typical mission includes a planning phase where the operator plans a route using a digital terrain data base. The operator then remotely drives the vehicle to a desired location as the vehicle records the route using navigation data. The operator activates the mission package for automatic target detection, and when targets are detected, the mission package designates them with a laser. The vehicle then automatically retraces the recorded route, a process termed *retro-traverse*.

In 1992 NIST demonstrated vision based autonomous driving, expanding its vehicle control applications into the civilian area as part of the Department of Transportation's Intelligent Vehicle/ Highway Systems (IVHS) program [4–9]. IVHS is a major initiative of government, industry, and academia to improve the Nation's surface transportation systems [10]. One IVHS component, the Advanced Vehicle Control System (AVCS), employs advanced sensor and control technologies to

assist the driver. In the long term, AVCS will provide fully automated vehicle/highway systems replacing the human driver altogether.

The use of vision-based perception techniques for autonomous driving is being investigated in many programs in the United States as well as in other countries [11]. Use of machine vision as a primary sensor has promise in that the infrastructure impact is minimized relative to other approaches.

This paper describes the testbed vehicle and support van. It presents the RCS reference model architecture for an autonomous vehicle and its implementation on the NIST vehicle. The paper then briefly describes the applications of teleoperation, retro-traverse, and autonomous driving.

2. TESTBED AND SUPPORT VEHICLES

The unmanned vehicle, a HMMWV, was actuated by NIST, ARL and the Tooele Army Depot as part of the DOD's Unmanned Ground Vehicles program [1,2]. Figure 1 is a photograph of the testbed vehicle. The vehicle contains electric motors for steering, brake, throttle, transmission, transfer case, and park brake and sensors to monitor the dashboard gauges indicating speed, RPM, and temperature.

Figure 1. Testbed vehicle followed by support van

A mobile computing and communications van was prepared to support NIST's development work [6,7]. This van houses development and support hardware, provides communication for operator control units during teleoperation, and contains the required computing systems to support lane following on public roadways. During lane following, video imagery is gathered by a camera on the HMMWV and is sent by a microwave link to the chase van. The image information is processed in the van. Vehicle control commands are computed and then sent back to the HMMWV control computer via an RF data link. Although the ultimate goal is to mount all vision processing and vehicle mobility controller real-time computational resources on the test vehicle, a portable development and performance evaluation facility will still be necessary.

3. RCS CONTROL ARCHITECTURE

One of the first steps performed by NIST to support its evaluation of autonomous vehicle component technology was to develop a reference model. The reference model describes what functions are to be performed and attempts to organize them based on a consistent set of guidelines [1,2].

Figure 2 shows a portion of the reference model architecture for an autonomous land vehicle. Modules in the hierarchy are shown with Sensor Processing (SP), World Modeling (WM) and Task Decomposition (TD). The sensory processing modules detect, filter, and correlate sensory

information. The world modeling modules estimate the state of the external world and make predictions and evaluations based on these estimates. The task decomposition modules implement real-time planning, control, and monitoring functions. The roles of these submodules are further described in [8]. This reference model has not been fully implemented but has served as a guide throughout the years as various control nodes were completed and as the vehicle's capability increased from teleoperation to autonomous driving.

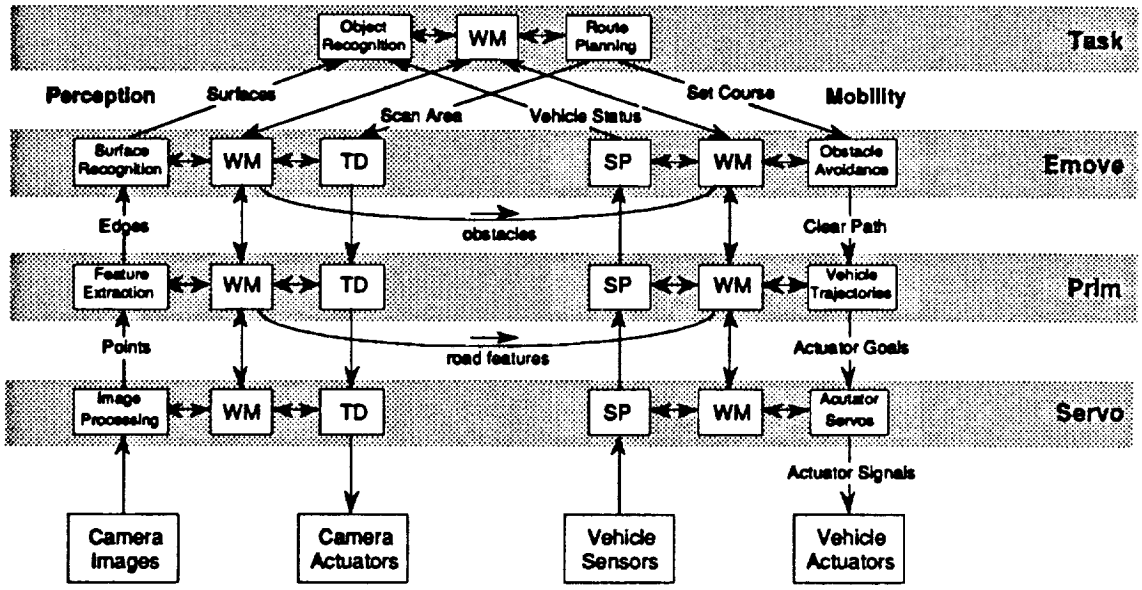


Figure 2. Reference model control architecture for an autonomous land vehicle.

The highest level of control for an individual vehicle, the Task level module, executes mission tasks phrased in symbolic terms, such as: Drive to exit 11 on I-270. A vehicle may be equipped with several subsystems, such as navigation, perception, and mission modules, which are directed by the Task level to achieve certain phases of the task.

The implementation for the U.S. Army Research Lab at Demo I used the lower two levels, Prim and Servo, of the mobility part of the reference model architecture to perform the mission elements. The servo level mobility controller drives motors for steering, brake, throttle, transmission, etc. and monitors the dashboard gauges. Vehicle navigation sensor data (position, velocity and acceleration) is processed and used to update the WM in the lowest level of the navigation subsystem. This data is used for steering and speed control of the vehicle during retro-traverse.

Extensions to the control system were necessary for implementing the IVHS autonomous road following [4, 5]. The lower two levels, Prim and Servo, on the perception side of the generic vehicle control system were developed. See Figure 2. The vision perception system uses a model of the lane edges to assist in the prediction and tracking of the lane markers on the road. The computed coordinates of the center of the lane are then used to steer the vehicle, in a similar fashion to retro-traverse.

Additional work on car following and collision avoidance requires the implementation of the next higher level of the control system, Emove. In this case the control system uses the visual surface features of the rear of the lead vehicle for lateral/longitudinal control in order to perform platooning [9]. Eventually, the performance of higher level tasks such as obstacle recognition/avoidance and route planning will require further extensions to the Emove and implementation of the Task levels of the architecture.

4. APPLICATIONS

Teleoperation

Although the ultimate goal for robotic vehicles is a fully autonomous system, control technology has not advanced far enough to realize this goal. Some form of operator intervention is needed, at least part of the time. For IVHS needs, the driver resumes control when the automatic system can not function. In a military setting the vehicle is unmanned and operator control requires some form of teleoperation.

The ARL vehicles communicate to a variety of operator control units. One is a small suitcase controller developed by NIST for field testing and is called the Mobility Control Station (MCS). A second operator station is housed in a tracked vehicle and is capable of controlling four unmanned vehicles at one time. This is called the Unmanned Ground Vehicle Control Testbed (UGVCT) and was developed by FMC for the Tank Automotive Command. Each system allows the operator to control all mobility functions. High level commands are issued using a touch screen display. A graphic display presents vehicle status to the operator.

Teleoperation is surprisingly difficult, hampered mostly, perhaps, by the difficulty in perceiving motion from a video image. To aid the operator, several areas are being investigated: force feedback, graphic overlays, and delay compensation.

Force feedback of the steering wheel provides the operator a feel for road conditions as well as sense of turn rate and vehicle speed. Unfortunately, closing a high speed force reflection loop places increased demands on an already burdened communication link. A simulated force feedback is being investigated. Here, vehicle speed and the operator wheel position is used to emulate the straightening torque that would be felt on the vehicle. The operator cannot feel the bumps in the road, but can get a sense of wheel position and vehicle speed. In addition, safety limits can be imposed so the wheel is not allowed to turn past a limit which is a function of speed.

In many situations, the operator can locate a clear path in the video but has trouble determining how much to turn the steering wheel in order to steer the vehicle over the clear path. To facilitate this, we are using a graphic overlay to represent the position of where the vehicle will travel at the given steering position. The projected vehicle position represented in the video assumes a flat ground plane and moves further ahead of the vehicle as forward speed increases.

Finally, we are investigating controller delay compensation. During teleoperation, several steps occur sequentially. The video camera takes an image, it is transmitted to the control station, the operator moves the steering wheel, the commanded wheel position is transmitted to the vehicle, and the actuator responds. Each step takes a finite amount of time, adding to the control delay. This delay can be very large especially for some forms of video compression. During this delay, the vehicle moves and the location of the desired path as specified by the steering angle changes position relative to the vehicle. Using navigation sensors, the change in position during the delay can be measured and the location of the desired path relative to the current vehicle position can be determined.

Retro-traverse

For retro-traverse, the vehicle's path is recorded during teleoperation allowing the vehicle to autonomously return along the path. During Demo I, this form of navigation allowed the vehicle to lay a smoke screen and travel through the smoke without the operator input. Driving through a smoke screen rules out the use of a vision system by a remote operator, but some form of obstacle detection is necessary in cases where vehicles or humans wander onto the path. A microwave sensor that would allow the vehicle to detect obstacles is being investigated.

The retro-traverse path is stored during the teach phase as a series of X-Y (or Northing-Easting) points. During the playback phase, a goal point is selected that is on the path and is a specified distance in front of the vehicle. The steering angle is computed using the "pure pursuit" method

[12]. The operator specifies the desired velocity and selects an automatic turnaround manuver. The Modular Azimuth Position System (MAPS), an inertial navigation unit, is used to sense vehicle position and orientation. MAPS uses ring-laser gyros and accelerometers to determine vehicle motion. An interface board (called the Navigation Interface Unit) and software to integrate vehicle odometry with MAPS data was developed by Alliant Tech and used during Demo I. Details of the navigation portion of the driving package are in [3].

Autonomous Driving

There are two low level functions required to drive a vehicle down the road, stay on the road and do not hit anything. NIST has been developing a vision based perception system to perform these functions.

The controller tracks the lane markers commonly painted on roadways and steers the vehicle along the center of the lane in the following steps. First, edges are extracted from the video image within a window of interest. Edges occur where the brightness of the image changes, such as where the image changes from a gray road to a white stripe. Then, quadratic curves that represent each of the two lane boundaries as they appears in the video image are updated. The system computes the coefficients of the curves using a recursive least square fit with exponential decay. The steering wheel angle that steers the vehicle along the center of the perceived lane is calculated using the pure pursuit method used for retro–traverse. Finally, navigation sensors compensate for the computation and transmission delay by adjusting the steering goal in accordance to the motion of the vehicle during the delay. More details of the vision processing and control algorithms can be found in [4,5].

Figure 3 shows the various scenes obtained when applying a window of interest to the road scene. The lateral position of the window of interest shifts in order to keep it centered on the road and its shape changes as a function of the predicted road curvature.

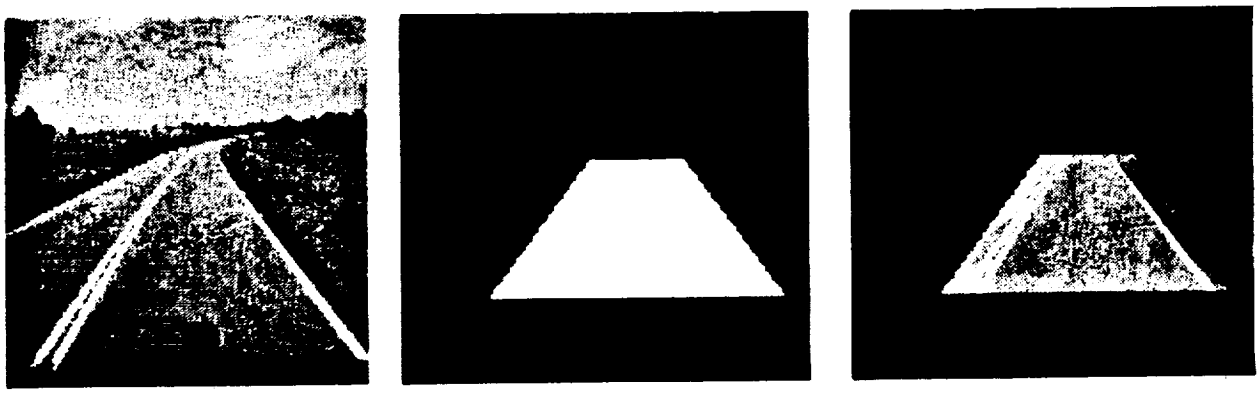


Figure 3. Road Scene, Window of Interest, Masked Road Scene.

The Montgomery County DOT permitted NIST to test the instrumented vehicle on a public highway. During these tests, autonomous driving was maintained over several kilometers (gaps in the lane markings at intersections prevented test runs of longer distances) and at speeds up to 90 Km/h. The vehicle has also been driven on various tests courses under weather conditions ranging from ideal to heavy rain, and under various outdoor lighting conditions including night time with headlights on.

Besides following the road, an autonomous vehicle must track and avoid obstacles and other vehicles. In addition, if the system can track another vehicle, it can follow that vehicle, forming a platoon. Platooning is envisioned by the military to reduce manpower requirements. In the IVHS version, vehicles would platoon at two meter spacings, in order to increase traffic throughput.

An approach to vision-based car following was developed that tracks the back of a lead vehicle or a target mounted on the back of the vehicle [9]. Since orientation is approximately constant during car following, the algorithm estimates only the relative translation of the lead vehicle. The system was tested using a video recording taken while the testbed vehicle was manually driven behind the lead vehicle. The system demonstrated tracking for vehicle separations of up to 15 meters.

5. SUMMARY

NIST's roles are to evaluate component technology for autonomous vehicles and to work with industry and academia to advance the state-of-the-art. To perform such a task, an architecture has been developed that will allow incremental development of autonomous capabilities in a modular fashion. The low levels of the control system have been implemented to support the DOD near term robotic tech base. That system was demonstrated at the 1992 Demo I. The control system was systematically extended to incorporate higher levels of autonomous capabilities to support further evaluations and developments in conjunction with the DOD tech base and DOT IVHS programs.

6. ACKNOWLEDGEMENTS

The authors would like to thank Chuck Shoemaker of the Army Research Laboratory and Richard Bishop of the IVHS Research Division at FHWA for their support and direction. Also, thanks to Roger V. Bostelman, Tom Wheatley, and Chuck Giauque of NIST for their help and dedication to the program.

REFERENCES

1. Szabo, S., Scott, H.A., Murphy, K.N., Legowik, S.A., Bostelman, R.V., "High-Level Mobility Controller for a Remotely Operated Unmanned Land Vehicle", Journal of Intelligent and Robotic Systems, Vol 5: 63-77,1992.
2. Szabo, S., Murphy, K., Scott, H., Legowik, S., Bostelman, R., "Control System Architecture for Unmanned Vehicle Systems, Huntsville AL, June 1992.
3. Murphy, K., " Navigation and Retro-traverse on a Remotely Operated Vehicle," IEEE Sigapore International Conference on Intelligent Control and Instrumentation, Singapore, February 1992.
4. Schneiderman, H., Nashman, M., "Visual Processing for Autonomous Driving," IEEE Workshop on Applications of Computer Vision, Palm Springs, CA., Nov 30-Dec 2, 1992.
5. Nashman, M., Scheiderman H., "Real-Time Visual Processing for Autonomous Driving" IEEE Intelligent Vehicles '93 conference, Tokyo, Japan July 14, 1993
6. M. Juberts, K. Murphy, M. Nashman, H. Scheiderman, H. Scott, S. Szabo, "Development And Test Results for a Vision-Based Approach to AVCS." 26th ISATA meeting on Advanced Transport Telematics / Intelligent Vehicle Highway Systems, Aachen, Germany. September 1993
7. Juberts, M., Raviv, D. "Vision Based Mobility Control for AVCS," Proceedings of the IVHS America 1993 Annual Mtg., Washington, D.C., April, 1993.
8. Albus, J., Juberts, M., Szabo, S., "A Reference Model Architecture for Intelligent Vehicle and Highway Systems", ISATA 25th Silver Jubilee International Symposium on Automotive Technology and automation. Florence, Italy, 1992.
9. Schneiderman, H., Nashman, M., Lumia, R., "Model-based Vision for Car Following." SPIE 2059 Sensor Fusion VI, Boston, MA September 8, 1993.
10. "Strategic Plan for Intelligent Vehicle Highway Systems in the United States," Report No. IVHS-AMER-92-3, published by IVHS America, May 20, 1992.
11. Schneiderman, H., Albus, J., "Survey of Visual-Based Methods for Autonomous Driving," Proceedings of the IVHS America 1993 Annual Mtg., Washington, D.C. April 1993.
12. Amidi, O., "Integrated Mobile Robot Control", M.S. Thesis, Carnegie Mellon University, May 1990.

N94- 34038

Intelligent Vehicle Control: Opportunities for Terrestrial-Space System Integration

Charles Shoemaker
Army Research Laboratory
Aberdeen Proving Ground, MD

For 11 years the Department of Defense has cooperated with a diverse array of other Federal agencies, including the National Institute of Standards and Technology, the Jet Propulsion Laboratory, and the Department of Energy, to develop robotics technology for unmanned ground systems. These activities have addressed control system architectures supporting sharing of tasks between the system operator and various automated subsystems, man-machine interfaces to intelligent vehicles systems, video compression supporting vehicle driving in low data rate digital communication environments, multiple simultaneous vehicle control by a single operator, path planning and retrace, and automated obstacle detection and avoidance subsystem. performance metrics and test facilities for robotic vehicles have been developed permitting objective performance assessment of a variety of operator-automated vehicle control regimes. Progress in these areas will be described in the context of robotic vehicle testbeds specifically developed for automated vehicle research. These initiatives, particularly as regards the data compression, task sharing, and automated mobility topics, also have relevance in the space environment. The intersection of technology development interests between these two communities will be discussed in this paper.

THE SERVICING AID TOOL:
A TELEOPERATED ROBOTICS SYSTEM FOR SPACE APPLICATIONS

N94-34039

Keith W. Dorman, John L. Pullen & William O. Keksz
Fairchild Space
20301 Century Blvd., Germantown, Md. 20874

James P. Karlen, Paul H. Eismann & Keith A. Kowalski
Robotics Research Corporation
P. O. Box 206, Amelia, Oh. 45102

ABSTRACT

The Servicing Aid Tool (SAT) is a teleoperated, force-reflecting manipulation system designed for use on NASA's Space Shuttle. The system will assist Extravehicular Activity (EVA) servicing of spacecraft such as the Hubble Space Telescope. The SAT stands out from other robotics development programs in that special attention has been given to provide a low-cost, space-qualified design which can easily and inexpensively be reconfigured and/or enhanced through the addition of existing NASA funded technology as that technology matures. SAT components are spaceflight adaptations of existing ground-based designs from Robotics Research Corporation (RRC), the leading supplier of robotics systems to the NASA and university research community in the United States. Fairchild Space is the prime contractor and provides the control electronics, safety system, system integration and qualification testing. The manipulator consists of a 6-DOF Slave Arm mounted on a 1-DOF Positioning Link in the shuttle payload bay. The Slave Arm is controlled via a highly similar, 6-DOF, force-reflecting Master Arm from Schilling Development, Inc. This work is being performed under contract to the Goddard Space Flight Center Code, Code 442, Hubble Space Telescope Flight Systems and Servicing Project.

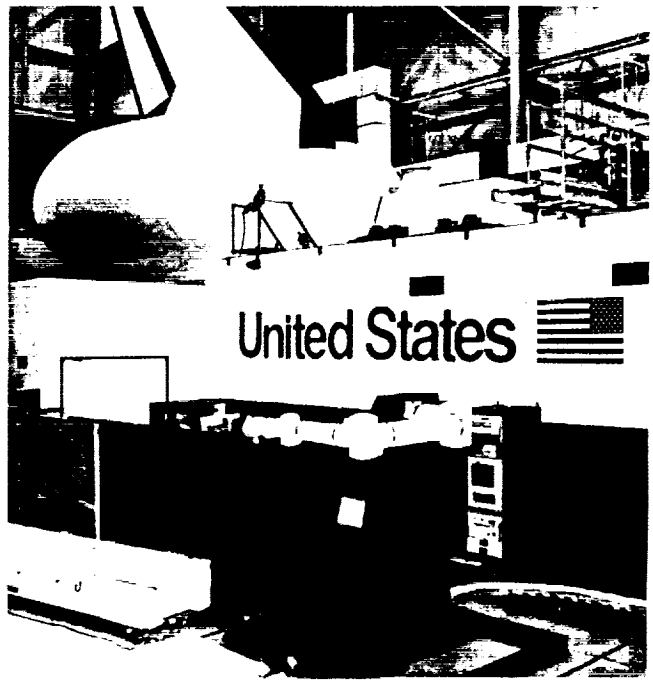
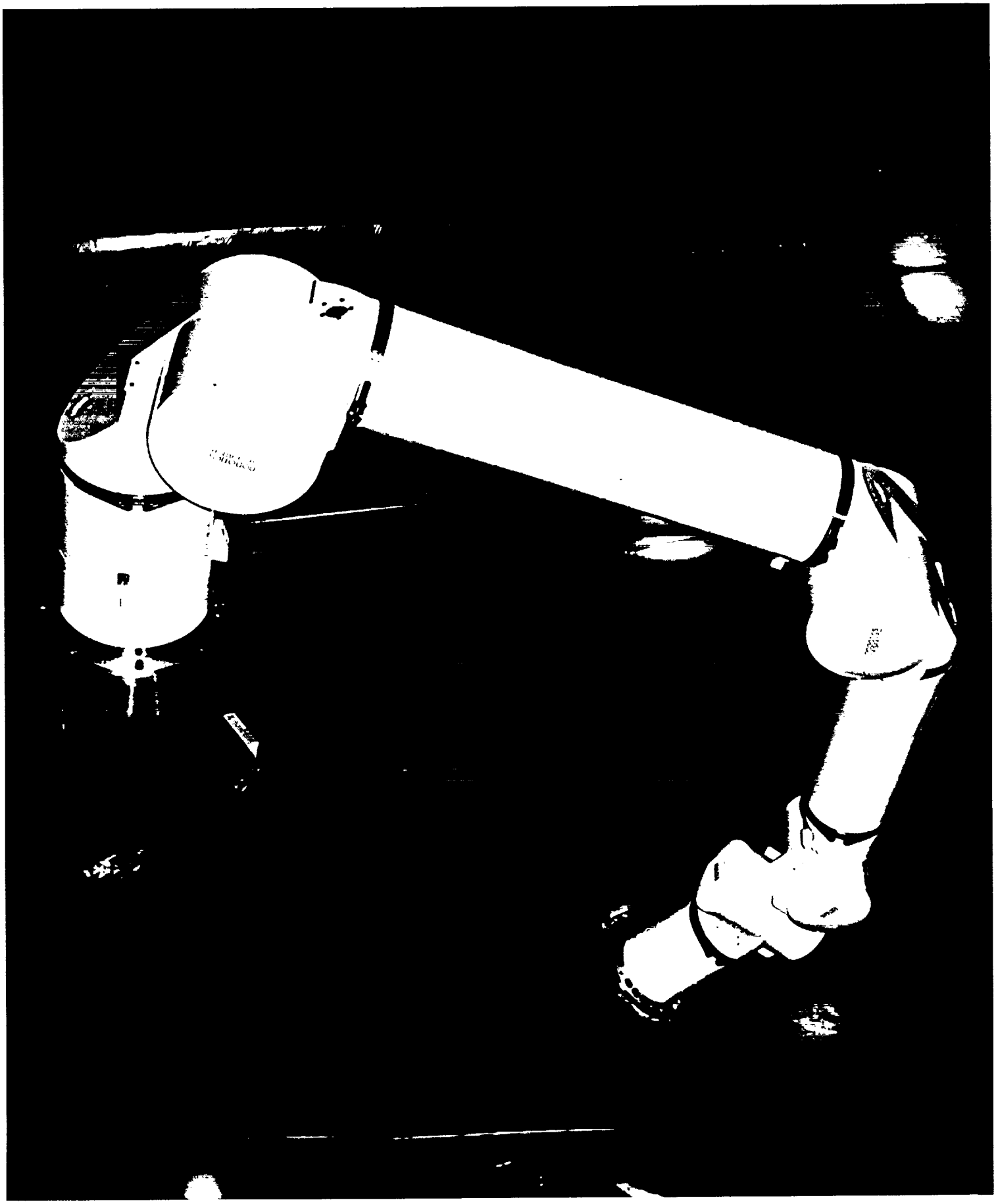


Figure 1. SAT Slave Arm at the GSFC

INTRODUCTION

In 1989, the Goddard Space Flight Center (GSFC) released a RFP for a low-cost, flight-capable, teleoperated robot system which could support 1G testing and training, and significantly improve on-orbit servicing of spacecraft. The subject robotics development program has been based on adaptations of existing robotics and military hardware, compatibility with existing and proven GSFC avionics used on the shuttle, slave arms directly descendant from the majority of robotics technology development platforms used throughout NASA and the universities, and designed ready to incorporate additional operational and controls features as may be required.



The SAT stands out from other robotics arms in the flexibility of its design to conform and adapt to changing needs with relatively little expense in doing so. Varying mission requirements and uncertain final requirements for safety compliance (anyone familiar with the safety review process knows that many failure mechanisms and corrective action requirements are not identified until the latter stages of the safety review process—not the Phase 0 or 1 levels) have received due consideration in the construction of the SAT. The SAT arm mechanism, shown in Figures 1 and 2, is composed of a series of self-contained joint drive modules joined by quick-disconnect band clamps Thus, it would be easy to re-configure the system to suit different user needs and applications. For instance, the current SAT Slave Arm has an 85 inch reach (shoulder centroid to toolplate). If determined to be advantageous for some particular flight application, the arm could be reduced to 60 inches in reach— or 48 inches or whatever dimension was appropriate— simply by shortening the hollow tubes which make up the forearm and upper arm segments. Alternatively, an additional joint could be added into one of these hollow tubes to provide increased dexterity as discussed latter in this paper.

Furthermore, the control computer has a substantial amount of growth capacity. Of 15 slots in the multibus chassis assembly, only 8 are currently used. Less than 10% of the bus bandwidth, and only 60% of the computational capacity is currently being utilized. Likewise, the companion electronics assembly to the control computer also has plenty of spare connector ports, relays, and power distribution to provide expansion.

Since the SAT is an operational 1G system it is the ideal candidate for technology transfer. Since their introduction in 1987, seven degree-of-freedom, position/force-controlled manipulators designed and manufactured by Robotics Research Corporation have served as the standard development platform across the NASA community for work in dexterous manipulation and space telerobotics. Users include the telerobotics laboratories at the Jet Propulsion Laboratory, Johnson Space Flight Center, Langley Research Center, Goddard Space Flight Center, the National Institute of Standards and Technology, Lockheed Engineering & Sciences Company, Lockheed Missiles & Space Company, Grumman Space & Electronics Group, Space Systems/ Loral, Fairchild Space & Defense Corporation, the University of Tennessee, Case Western Reserve University and NEC (Japan). As a consequence, a considerable body of advanced control technology compatible with these products, as well as in-depth application and integration experience, now exists.

At least 39 separate research and development projects have been undertaken by researchers in this community to date, 29 of which were conducted at NASA and NIST since 1987 (including 10 current NASA projects) and the remainder at academic institutions and research oriented companies.

New technology developed in these projects include alternative approaches to kinematics for 7-DOF manipulators, high bandwidth force control software using the internal joint torque sensors provided in RRC arms, calibration techniques for redundant arms, evaluations of alternate hand controllers and user interfaces, and architectures for high-level autonomous and supervisory control systems. Applications demonstrated to date include Space Station inspection, Space Station truss assembly, satellite servicing tasks, on-orbit assembly of aero brakes, simulation of spacecraft docking mechanisms and the development of robot-friendly truss fasteners.

Recently, several large U. S. industrial corporations have begun seriously evaluating the use of RRC type manipulators for factory use. In this light, the SAT offers an excellent vehicle by which to implement NASA-funded technology toward improved national competitiveness.

SYSTEM DESCRIPTION

The Servicing Aid Tool (SAT) is designed to allow an Operator to control a teleoperated six degree of freedom Slave Arm using a six degree of freedom, force-reflecting Master Arm. The master and slave arms have highly similar kinematic arrangements, both being configured in the same manner: a roll/pitch shoulder, a pitch elbow, and a pitch/yaw/roll wrist.

This allows use of a joint-to-joint control scheme: a joint on the Slave Arm is commanded by motion of only the corresponding Master Arm joint, and a torque signal is provided to each Master Arm joint as a result of the state of the corresponding Slave joint. Force commands are reflected to each master joint based on the corresponding slave joint torque sensor. The torque sensor also provides feedback for a local analog torque loop which eliminates the effect of friction in the joint.

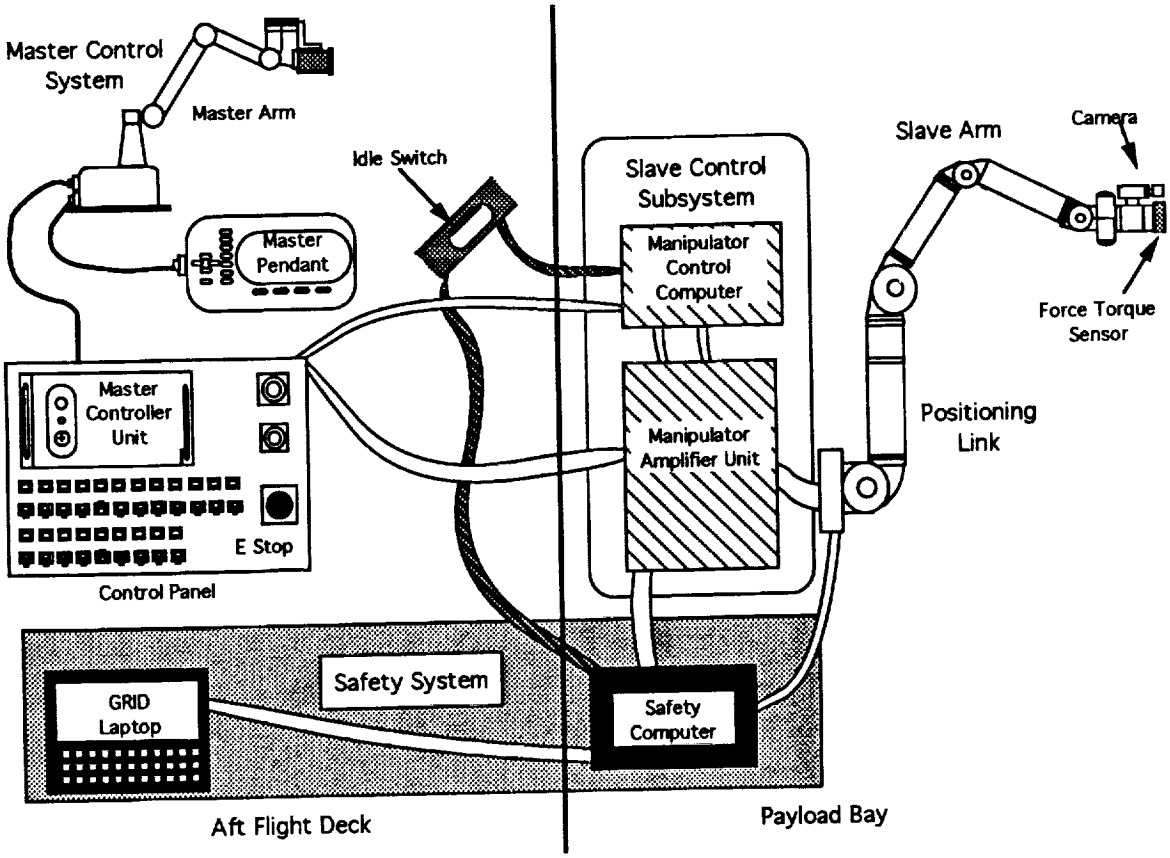


Figure 3 SAT Subsystems

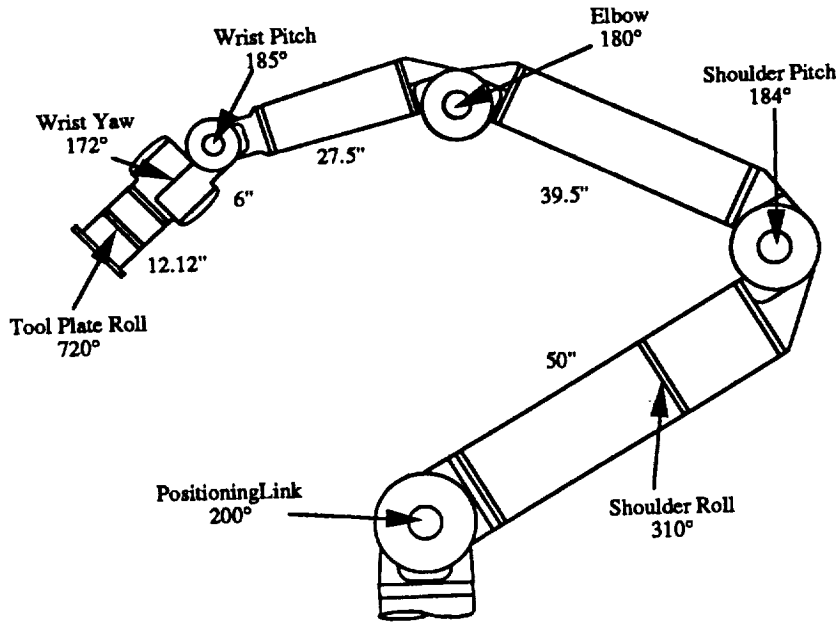


Figure 4. SA/PL Dimensions and Joint Travel

The one degree of freedom Positioning Link is controlled via operator interface keyboard commands, and operates only when the Slave Arm is disabled.

The kinematics are simple, with the three adjacent pitch joints allowing the Operator to mentally separate the position and attitude of the tool: the shoulder and elbow joints provide position; the wrist joints, attitude.

The SAT components (Figure 3) are spaceflight adaptations of existing ground-based designs. The Master Arm is a slightly modified Schilling Development OMEGA from the Titan 7F master/slave system used in undersea systems. The Slave Arm and Positioning Link (SA/PL) are configured to mimic the Schilling Titan 7F Slave Arm kinematics.

To increase the functionality of the SAT, it will be relocatable via the Shuttle Remote Manipulator System (RMS) to various worksite locations where Hot Shoe receptacles are stationed. The hot shoe will provide a releasable electrical and mechanical interface, allowing the SA/PL to be moved to another location, or to be jettisoned in an emergency. A Grapple Fixture will be provided to allow the Shuttle RMS to move the SA/PL. Remote release will be single-fault-tolerant and commanded from the Aft Flight Deck, backup release may also be performed manually via EVA. Inadvertent release will also be two-fault tolerant. The low replacement cost of the slave arm combined with the jettison capability provide a cost-effective means of compliance to the safety requirements for two fault tolerance.

SPACE QUALIFICATION

The SAT components will undergo environmental testing (vibration, thermal/vacuum, and EMI) at protoflight levels. Where necessary, modifications have been made to upgrade designs to protoflight levels. The primary effect has been on the electronics. The RRC Multibus boards in the control computer, for example, had to be replaced with military versions packaged to survive the vibration and thermal environment. A similar version of our protoflight control computer successfully flew on the shuttle for the TSS program. There have also been design changes in the RRC manipulator components to meet outgassing, venting, thermal, and fracture control requirements.

PAYLOAD BAY COMPONENTS

Slave Arm and Positioning Link

The SA/PL dimensions and joint travel are shown in Figure 4. Figure 5 illustrates the layout on the Flight Support System (FSS), a cross-bay carrier intended for supporting large spacecraft. Components in the Payload Bay are listed below.

All Slave Arm joints have brushless DC motors, operating through a 160:1 harmonic drive. The joint output side is connected through a hollow shaft to a resolver, which reads the angle between the two adjacent links, rather than motor driveshaft angle. In like manner, the strain gauges are mounted to read the output torque of the joint, being mounted at the base of the harmonic drive. Both sensors thus measure the true relationship between the input and output sides of the joint, eliminating the effects of friction and any cogging of the harmonic drives.

The travel for each joint is limited, in order, by software limits, limit switches, and hard stops. Passing a limit switch results in removal of power from the motors and brakes, thus engaging the brakes. The brakes may be remotely disengaged from the Aft Flight Deck (AFD) control panel without powering the motors to allow EVA stowing as a backup.

The SAT is designed to demonstrate its capabilities on the ground as well as to perform on orbit. It is capable of lifting a 20 lb mass in a 1-G environment at any pose within its range of joint travel. The design point for the 0-G case is for a 500 lbm payload.

To provide an interface for an exchange mechanism, tool, and camera, the Slave Arm is designed to be compatible with a variety of exchange mechanisms; it will provide power and data for operation of the exchange mechanism, tool, and camera. The exchange mechanism will be two-fault tolerant to ensure the ability to release tools and ORUs and stow the arm. Several mechanisms are currently under evaluation. Tools will be specified as part of the mission integration in a future program phase.

The maximum joint rates are specified so that no single joint runaway can cause a tool plate velocity in excess of 17 inches per second; this value was chosen as typical of RMS maximum rates.

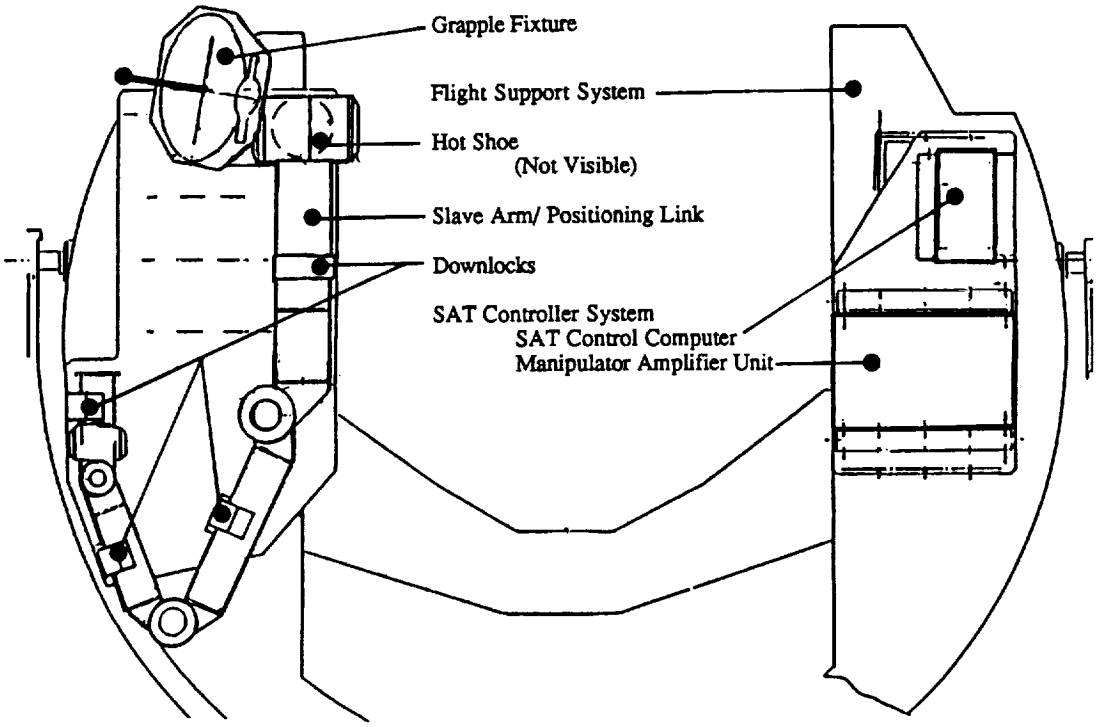
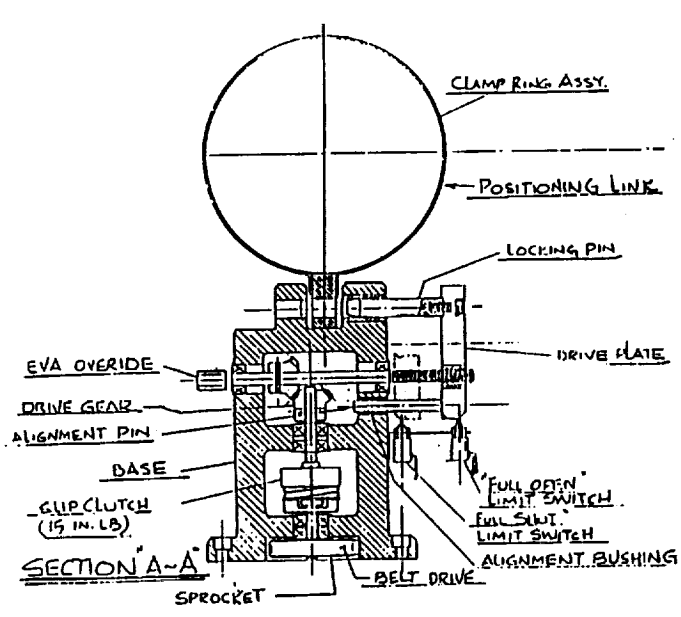
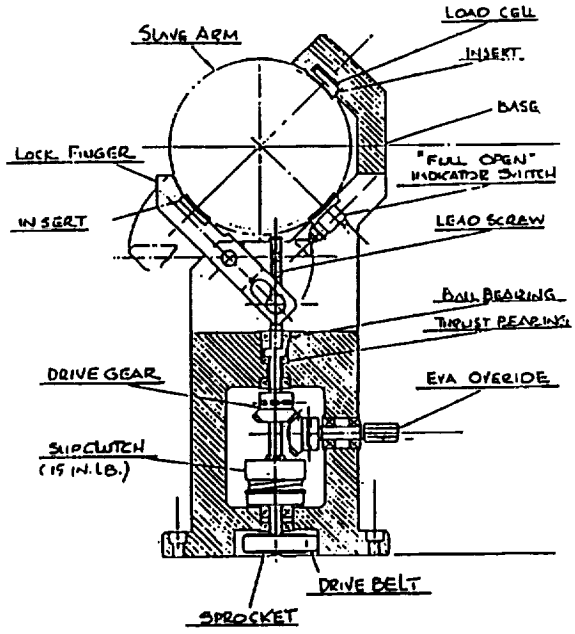


Figure 5. SAT Components Mounted on a Cross-bay Carrier



Positioning Link



Slave Arm (1 of 3)

Figure 6. Prototype Downlocks

Slave Mounting Assembly

The Slave Mounting Assembly is the means by which the SA/PL is mounted to its cross-bay carrier, and includes a Mounting Plate, Downlock Mechanisms, Hot Shoe, and Grapple-Hot Shoe Adapter Plate (GHAP).

The Downlocks secure the SA/PL for launch and landing. There is a downlock for each of the four SA/PL links - three for the SA, one for the PL. Figure 6 depicts the prototype downlock design that is to be used both for demonstration and vibration testing; these will be driven via a power wrench. The protoflight downlocks will be driven by a standard FSS Common Drive Unit, and will incorporate load sensors and limit switches to stop power to the drive unit when sufficient torque is read; slip clutches will limit forces on each SA link. Redundant sensors will be incorporated to reliably indicate that the SA/PL is positioned to allow closing the downlock, and that the SA/PL is positively locked after actuation.

Slave Controller Subsystem

The Slave Controller Subsystem (SCS) provides the interface between the master and slave systems, and the control engine and power for the SA/PL. There are two components, the Manipulator Control Computer (MCC), and the Manipulator Amplifier Unit (MAU). These are mounted on a radiator plate, which is in turn mounted on the cross-bay carrier. Both units will be subjected to the appropriate environmental testing for space qualification.

The MCC contains two 80386 based processors for SA/PL control and Master Arm force command generation and another 80386 for communications with the MCS. Slave arm data acquisition is accomplished via MCC resident A/D, D/A, and R/D (resolver to digital) hardware. The MAU contains the motor amplifiers and an analog torque loop compensator for the SA/PL actuators, and watchdog electronics which check the health of the MCC processor boards and secondary power. There are a total of 8 amplifiers, one of which is a back-up which may be switched to any individual joint for manually-controlled operation of a joint.

The system is equipped with an Emergency Stop Current Loop which, when broken, will cause the Slave Arm and Positioning Link to become disabled. The Emergency Stop Current Loop can be broken by Operator action, software command or hardware command. The current loop nodes are shown in Figure 7. Each node is actually a current pass-through which can be broken by the shown input.

AFT FLIGHT DECK COMPONENTS

Master Controller Subsystem

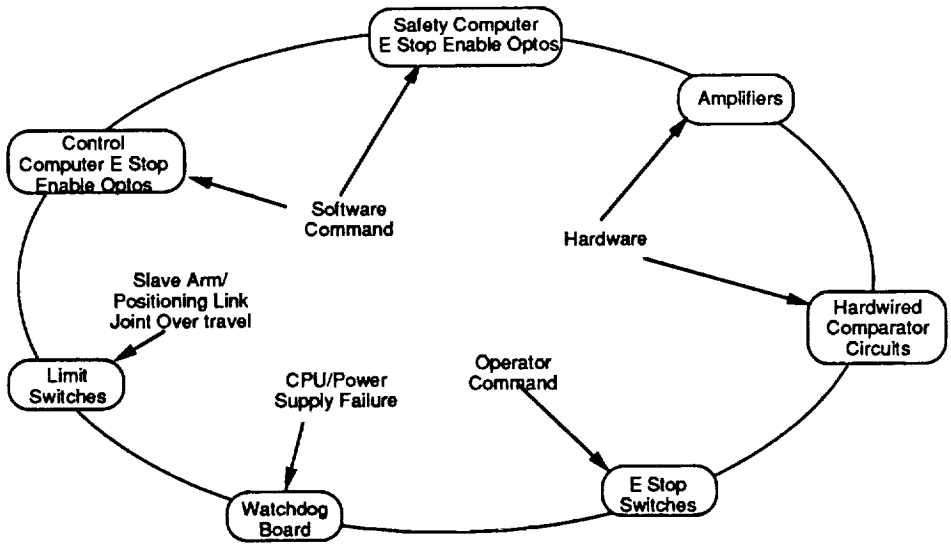
The Master Controller Subsystem consists of the modified Schilling components (Figure 8)- Master Arm with a reach of 16 inches, Master Pendant, and Master Control Unit. The Master Arm and pendant are mounted on the master Mounting Assembly; The MCU is inserted into the Control Panel. The MCS components are stowed in a mid-deck locker for launch and landing, packed in a foam material for protection from the loads.

Control Panel and Master Mounting Assembly

The MCS and Control Panel provide the Operator complete control of the system. The Control Panel, mounted in the L11 panel (Figure 8) has control switches for the SA and PL power enables; an Emergency Stop (E-Stop) button, which cuts power to the joint actuators and engages the brakes; and joint brake and limit switch overrides. The latter, in conjunction with controls for a backup single-joint means of operation, allow recovery from some fault conditions which would otherwise cause the Slave Arm to "freeze," preventing stowing.

The L11 panel also provides connections for the Idle Switch, incorporated into a mounting bar attached in the vicinity of the control panel. The Idle Switch is placed so that it provides a stabilizing grasp point for the Operator to react against the Master Arm torques (additional stabilization will be provided by foot straps on the AFD floor). The bar is positioned to allow view of the AFD monitors, as well as a view out the AFD windows, and is designed to allow mounting the master operator interface as well as other tool controls within easy reach of the operator.

In order for the Slave Arm to move, the Operator must depress the Idle Switch on the mounting bar. Releasing the Idle Switch while Slave Arm or Positioning Link motion is being commanded will cause the Slave Arm to decelerate and stop. Motors are not disabled but master and Slave Arm joints are servoed to their current



The Emergency Stop Current Loop is a continous current loop, which when interrupted causes the slave arm and positioning link to become disabled. The current loop can be interrupted by any of the nodes in the current loop.

Figure 7. Emergency Stop Current Loop Nodes

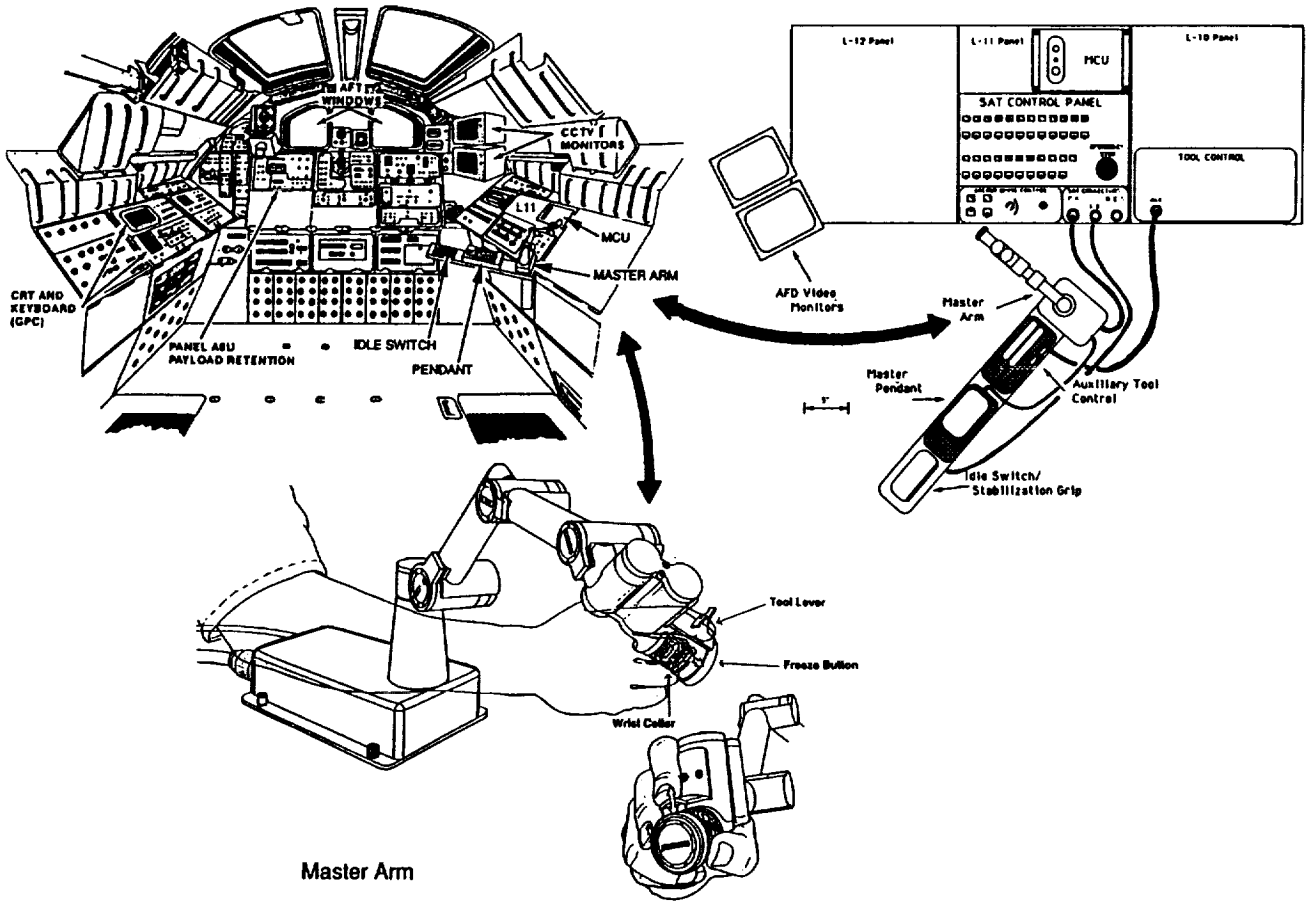


Figure 8. Aft Flight Deck Installation

position. The master and slave arms will maintain their position until the Idle Switch is pressed and the arm is commanded to move again.

SAFETY ANALYSES AND CONTROLS

In December 1991, a Technical Interchange Meeting was held with the JSC Payload Safety Review Panel (PSRP). Following some design changes a Phase 0 Safety Review was held in June 1992. The June review was intended to be a Phase 0/1 review of the SAT protoflight hardware and the level of detail for this hardware was commensurate to the Phase 1 level. However, the PSRP argued that since the tasks and ancillary tools not under contract were not well defined, the review would only count as a Phase 0. Following the review, the PSRP chairman commended the technical approach, and proclaimed that we were exceptionally forthcoming with possible fault mechanisms and creative solutions as inhibits.

A Structural Assessment and Hazard Analysis was performed for the SAT to ensure that neither normal operation nor dual failures could result in hazards to the Orbiter, crew, or other critical hardware. To perform these analyses, each subsystem was initially reviewed for its potential to create hazardous functions or effects. The review considered the subsystem design, materials, functions, and interfaces to other subsystems. This section describes the various hazard groups that were considered and the controls against them.

Aft Flight Deck Hazards

The fault tree analysis identified hazard causes within the aft flight deck since the Master Arm and the control panel are used there to operate the system. The Master Arm and control panel used on the aft flight deck can pose hazards to the crew. A mechanical hazard would be uncontrolled motion of the Master Arm; however, as the Master Arm is capable of exerting a maximum of only two pounds force, any injury would be minor.

EVA Hazards

The SAT is not presently planned to be powered during EVA operations. There are also no procedures that require astronaut intervention to return the payload bay to a safe condition except as a third control (inhibit) to removing the SAT from the bay in the event of a non-operating SAT failure where the SAT obstructs the bay doors or is failed in a position unsafe for landing.

Inability to Stow the SA/PL

If the SAT fails such that it cannot be commanded to its stow position, it could prevent closing the Payload Bay doors, or be unable to withstand the forces of re-entry and landing. In this case, the first option is to use the single-joint backup drive. The second is to disengage the joint brakes to allow an EVA crewmember to manually stow the SA/PL. This can be commanded by overrides available at the Control Panel. These cause power to be applied to the brakes but not to the actuators. An EVA crew member can then manually drive the SA/PL into the downlocks, while the override switches are held down by the Operator. The brakes and downlocks may then be engaged from the Control Panel.

If this proves to be impossible in the available time, the SA/PL may be jettisoned via command from the AFD to release the Hot Shoe. Depending on the Hot Shoe design chosen, jettison may be self-actuated, or may require the RMS to bring the SA/PL out of the Payload Bay. Remote release of the Hot Shoe will be redundant, the Hot Shoe will also provide for release via EVA should remote release fail.

Impact During Operation

Unplanned impacts during operation could cause damage to the orbiter, payloads, or SAT. Such impacts could be caused by failure of the SAT control system, sensors, or actuators; or by Operator error. The SAT system incorporates inhibits against such failures.

The maximum single-joint runaway rates produced by SAT are specified to minimize the possibility of damage to the Orbiter or payloads, and are comparable to those produced by the RMS; they are not optimized for a particular mission. Furthermore, Operator-adjustable limits are incorporated in software in the SCS, commandable via the master operator interface.

If the Operator suspects abnormal operation, he will first release the Idle Switch, which will result in a controlled stop for most faults. The Operator and/or the Monitor may also hit their respective E-Stops, which will shut down all power to the SA/PL, engaging the brakes.

Safety System

The SAT will also have a Safety Computer nearly identical to the MCC. It will monitor SAT's performance and shut down the system in the event certain parameters (Torque, joint rate, etc.) are exceeded. Some of these tests are redundant with those internal to the control computer. The Safety Computer interfaces directly to the Slave Arm analog feedback and control signals, rather than relying on data processes by the Control Computer; this reduces the chance that a computer fault might mask a fault elsewhere in the system. Additional features being considered include:

— Use of a toolplate force/torque sensor
— Incorporation of proximity sensors distributed along the SA/PL.
— World models of the Orbiter and Payload Bay to establish stay-out zones and automatic reduction in torques and rates when in proximity operations.

The SAT also incorporates independent hardwired adjustable limit-setting hardware. During operation, this hardware operates independent of all system computers, so is not susceptible to any computer faults. When any pre-set limit is exceeded, the SA/PL is disabled.

After operation has been completed the Slave Arm can be disabled by entering a disable command via the operator interface. The Slave Arm can also be disabled using the Emergency Stop Switch, however, it is primarily intended to be used when a quick shutdown is required.

SYSTEM OPERATING MODES

The SAT software operates in the following modes, which are commandable by the Operator via the master operator interface in the aft flight deck.

System Mode

The software enters the System mode when powered up, and it may be re-entered by command from the master operator interface, or by an E-Stop commanded by an Operator or by safety software. This mode allows health checks to be performed, and is the only mode that allows parameter updates. No SA/PL motion can occur, as it is unpowered, with brakes engaged.

Idle Mode

The Master and Slave Arms servo to current positions, with brakes disengaged; no commanded motion is possible. This mode is first entered when commanded from the System Mode. The other modes may then be commanded, but will not be entered until the Idle Switch is depressed. It is re-entered when the Idle Switch is released.

Teleoperation Mode

This is, of course, the mode in which most of SAT's work will be done. The Slave Arm responds to commands from the Master Arm. On transition into and out of this mode, both master and slave torques are ramped up and down to prevent step inputs to the worksite and to the operator. Scaled (slave rate less than master rate) or unscaled motion may be chosen via the operator interface. Indexed operation may be initiated by releasing the Idle Switch, moving the Master Arm to a new reference position, and then re-gripping the Idle Switch. These features have been found useful for fine control in proximity to or in contact with the worksite, and provide a flexible means of matching the Slave Arm to the Operator and to the needed task.

Automated Task Mode

A limited number of automated moves will be possible, and are commanded by keyboard input to the Operator interface. These operations still require the Idle switch to be depressed for motion to occur.

— Auto Stow/Unstow -
SA/PL commanded into and out of the downlocks
— Master to Slave Align -
Master assumes current pose of Slave Arm
— Slave to Master Align -
Slave assumes current pose of Master Arm
— Slave to Commanded Position -
Joint angle values input via operator interface
— Positioning Link is always commanded via Operator interface

Backup Single-Joint Mode

In addition to the above modes, which all require software, there is a backup Single-Joint Drive mode available, which is commanded completely via the control panel. A rotary switch is used to choose which joint is to be driven by a separate servo amplifier; another switch controls direction, and a knob the rate.

Powerup and Shutdown Operation

The MCS is powered up via the MCS Power Switch. After the MCS has initialized itself (as indicated on the MCS operator interface screen) the SCS, SA and PL can be powered up. The SCS, Slave Arm and Positioning Link are powered via the appropriate Control Panel power switches.

After the SCS has been powered it performs a self test and checks the status of the Slave Arm and Positioning Link. It communicates all status information to the operator interface. If everything passes, the Operator must verify all operational parameters. Among the status information checked are joint torque, position, temperature and limit switch status.

After all parameters have been verified, the Slave Arm can be enabled. To accomplish this, first the Emergency Stop System must be activated by pressing the Enable E Stop Switch. Next, the Slave Arm can be enabled by entering an enable command via the operator interface then pressing the enable switch on the Control Panel.

After operation has been completed the Slave Arm can be disabled by entering a disable command via the operator interface. The Slave Arm can also be disabled using the Emergency Stop Switch, however, it is primarily intended to be used when a quick shutdown is required. Note that the Idle Switch stops motion, but does not disable the arm.

POTENTIAL ENHANCEMENTS USING EXISTING TECHNOLOGY

Since the flight-qualified Servicing Aid Tool (SAT) mechanism and its control system are functionally identical to NASA's RRC laboratory units, many of the technologies that have been developed by NASA can be applied directly to the SAT to increase its capabilities for satellite servicing with minimum risk and expense. Five specific enhancements being considered are listed, as follows, in proposed order of implementation:

1. Addition of a High-Level Telerobotic Control System

One of several available versions of a high-level telerobotic control system (JSC, GSFC, JPL) could be implemented on new computer boards added to the existing SAT control system to provide programmable operation, 6-DOF kinematic cartesian control (i.e., the ability to command straight line moves) and a more powerful user interface. Space for such additional boards is already provided in the current SAT control hardware arrangement.

2. Addition and Evaluation of Alternative Hand Controllers

The Schilling replica master force-reflecting hand controller currently used in the SAT system is but one of several alternatives available. With the implementation of the above-described high-level controller and 6-DOF kinematics, two other types of hand controls which could offer advantages in certain SAT operations and may be preferred by the astronaut users can easily be interfaced and compared. Specifically, it is felt that a pair of standard 3-DOF rate controllers should be tried (as used to operate the RMS today), along with a 6-DOF hybrid rate/force controller from Cybernet Systems. Both types of hand controller have already been procured by NASA and could be made available. In general, it is anticipated that the ability to perform straight line moves with a rate controller—essentially to "fly the hand" of the SAT—will greatly simplify certain teleoperated tasks like extracting ORUs.

3. Addition of Impedance Control Software

Implementing existing impedance control software on the SAT will give the operator the ability to regulate electronically the apparent stiffness of the manipulator arm as it executes a contact operation. Essentially, this feature will permit the manipulator to control the forces and moments it exerts when mating two rigid parts (as in ORU insertion). Impedance control is particularly advantageous when using a rate controller to perform contact operations, since tool/workpiece reaction forces can be controlled (and limited) with great accuracy.

4. Addition of 6+1-DOF Kinematics

A 7-jointed manipulator arm affords an infinite number of arm postures for any given position and orientation of the tool (and the payload). Like the human arm, it can thus work around objects in the work space without collisions, providing significantly more capability to perform complicated manipulation tasks in cluttered environments. The current SAT slave arm has six degrees of freedom (one joint is also provided on the positioner link that supports the slave arm). To increase dexterity, it is recommended that a seventh joint be

added to the slave arm (an "elbow roll" joint), giving the operator the ability to change the elbow orientation, as a separately controlled joint, during operations. This new seventh joint would only be used, in this case, for arm reconfiguration and would not be active during the execution of tool-handling tasks. Once the operator has selected a preferred elbow posture, the slave arm would be controlled as a 6-DOF system.

5. Addition of Redundant 7-DOF Kinematics

With no further changes to the 6+1-DOF slave arm mechanism beyond those described above, more powerful redundant control software could be added to the SAT system if a prospective servicing application demands the enhanced capabilities afforded by active redundancy. Benefits include proximity sensor-driven, reflexive collision avoidance, by which the arm automatically changes its posture to avoid collisions with objects in the workspace, and automatic selection of the optimal arm pose to avoid singularities and improve leverage.

PROGRAM STATUS & CONCLUSION

The protoflight slave arm and controller are currently undergoing verification testing at Robotics Research Corporation. This hardware is due to ship to the GSFC by mid-August. Upon delivery, the master/slave communications software, gravity model, and force feedback software will be ported over to the protoflight controller for integration of the full-up master/slave system. The protoflight system will then proceed to environmental testing expected to be completed around the end of the calendar year. In January 1994, the basic SAT will be qualified for the rigors of space flight.

Future phases of the program are anticipated to continue ground demonstrations and to include the incorporation of selected enhancements. These enhancements will primarily be chosen to best augment the SAT's capabilities to perform a range of servicing tasks directed toward the second Hubble Space Telescope (HST) servicing mission. Current mission analyses for the first servicing mission support the postulate that the SAT will enhance astronaut tasks and timelines. The Servicing Aid Tool will provide a telerobotic complement to significantly enhance extravehicular capabilities.

Reference

Pullen, J. L., et al, "The Servicing Aid Tool," Cooperative Intelligent Robotics in Space III, Boston MA, November 15-20, 1992

N94- 34040

Robotic Vehicle Mobility and Task Performance – A Flexible Control Modality for Manned Systems

Frederick Eldredge
Tooele Army Depot
Tooele, UT

In the early 1980s, a number of concepts were developed applying robotics to ground systems. The majority of these early application concepts envisioned robotics technology embedded in dedicated unmanned systems; i.e., unmanned systems with no provision for direct manned control of the platform. Although these concepts offered advantages peculiar to platforms designed from the outset exclusively for unmanned operation – i.e., no crew compartment – their findings would require costs and support for a new class of unmanned systems. The current era of reduced budgets and increasing focus on rapid force projection has created new opportunities to examine the value of an alternative concept: the use of existing manned platforms with an ability to quickly shift from normal manned operation to unmanned should a particularly hazardous situation arise.

The author of this paper addresses the evolution of robotic vehicle concepts and technology testbeds from exclusively unmanned systems to a variety of "optionally manned" systems which have been designed with minimum intrusion actuator and control equipment to minimize degradation of vehicle performance in manned modes of operation.

SECTION II

AUTOMATION AND INTELLIGENT SYSTEMS

Session A1: ARTIFICIAL INTELLIGENCE I

Session Chair: Dr. Peter Friedland

Artificial Intelligence at AFOSR

Dr. Abraham Waksman

202-767-5028
DSN: 297-5028
FAX: 202-404-7496
waksman@isi.edu

The Challenge

To **scale up** with real world data and problems;
To operate effectively with **uncertain information**;
To introduce **adaptability** and **learning**
so that systems will improve with use;
To operate effectively in a **resource bound environment.**

1. **The portfolio:** Program mix and rationale

2. **Imaging:** Machine vision/image understanding: visual, IR, RF bands

3. **Information Management:**
   Cooperative DB, Anytime computation, CBR

4. **Tools:** Learning, Numeric Methods, Uncertainty,
   Resource bound computing

5. **Example of the Science:** Theory of Invariants applied to Imaging

The Portfolio

Need to optimize return on investment

Existing programs are judged by progress toward transition (income).

New projects are judged by the promise
(growth, technical/scientific breakthrough).

30% new starts/year are desired, 15% new starts realized

Program Manager is accountable for the
relevance of the program and its components.

Program Manager is expected to promote the science of interest
within the scientific community.

Imaging

Air Force needs: Image interpretation in the nonvisual range for ATR; Image interpretation in the visual range—machine vision/ image understanding for Mission Rehearsal and Training; Autonomous vehicle navigation (space)

Activity of program participants:

Model-based object recognition: Feature extraction for models, feature decomposition/composition for hierarchical model bases
Theory of Invariants: Geometric invariants obviate pose and scale, and drastically reduce the number of models needed. Algebraic invariants reduce complexity of computation in 2D to 3D transformation for recognition.
Adaptation/learning: active vision, motion, inductive clustering techniques
Cartography: Automatic knowledge acquisition for recognition of targets in aerial surveillance, mapping, navigation
Model-based ATR: Exploit model-based vision (MBV) concepts, apply them to the nonvisible band (IR, RF)

Information Systems

Air Force needs:

Management of large knowledge/data-based systems,
Retrieval and update
Decision aids which are timely

Activity of program participants:

Anytime algorithms, monotone retrieval, nearest neighbor methods (CBR, Bayesian), formal structures for independent agents communication, Deductive Databases, cooperative databases, Visualization, adaptability in DB management, knowledge sharing

Tools and Techniques

Learning: Introduce inductive processes in vision, knowledge/data organization

Integration of numeric and symbolic computing: Merge inference techniques of symbolic progressing (Deductive, Inductive) with inference techniques of numeric processing (probability and statistics), with OR techniques (linear programming)

Uncertainty: Integration of Belief Systems (based on logic) and Bayesian methods (statistically based) to effectively utilize all available knowledge

Resource Bound Computing: "Satisficing"— computing under time constraints and incomplete information

SAR Imaging

SAR instruments permit high-resolution imaging from space.

SAR imaging is possible in all weather conditions.

The Problem:

Poor resolution makes SAR images difficult to interpret.
There is no shading or shadow information,
and appearance of objects changes radically with pose.

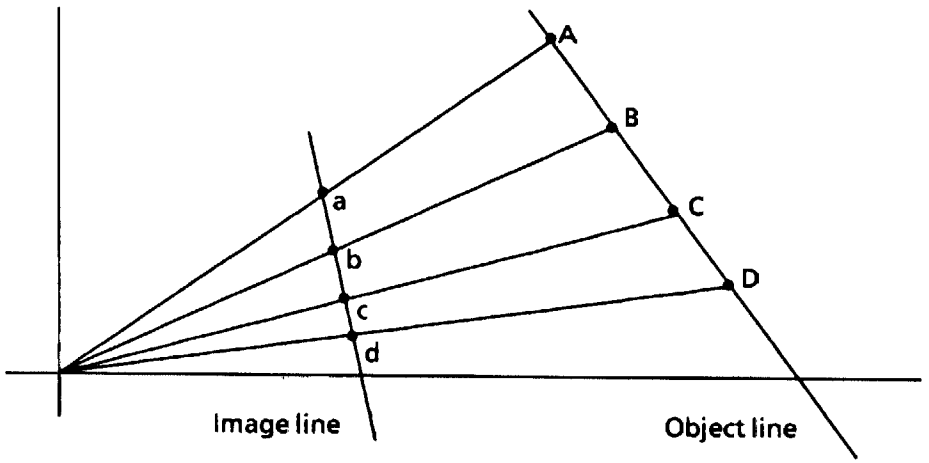
The Approach:

Projective invariants and model-based object recognition techniques

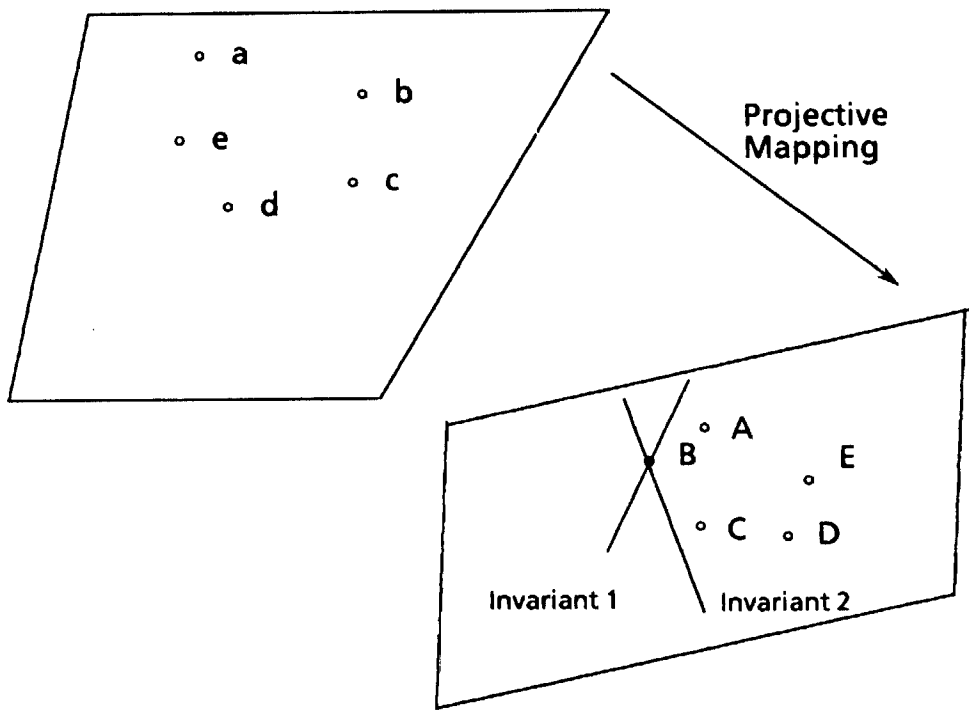
Projective Invariants

The objective of finding projective invariants is recognizing the same objects in images that differ in scale, tilt, and rotation,

Given collinear points A B C D,
the value ((AB)(CD)) / ((AC)(BD)) is invariant to projection.
This invariant is the **cross-ratio** of four collinear points.



The cross-ratio of distances between any four points in the object line is the same as the cross-ratio of the distances between their images in **any** image line.



Transfer of a point in two dimensions using the five-point invariance. Two invariants define a pair of linear constraints which define a point.

Summary

Invariants open the door to efficient image interpretation

OR techniques in AI open the door to efficient search within symbolic computation (resolution and unification)

Decision theoretic approaches to belief and probability open the door to more complete information utilization (possibility and probability)

All are integrations of mathematics (numeric) and AI (symbolic)

All will contribute towards **scale-up, robustness, and efficiency.**

N94-34041

The Stanford How Things Work Project[1]

Richard Fikes, Tom Gruber, and Yumi Iwasaki

Knowledge Systems Laboratory
Stanford University
701 Welch Road, Building C
Palo Alto, CA 94304

Abstract

We provide an overview of the Stanford How Things Work (HTW) project, an ongoing integrated collection of research activities in the Knowledge Systems Laboratory at Stanford University. The project is developing technology for representing knowledge about engineered devices in a form that enables the knowledge to be used in multiple systems for multiple reasoning tasks, and reasoning methods that enable the represented knowledge to be effectively applied to the performance of the core engineering task of simulating and analyzing device behavior. The central new capabilities currently being developed in the project are automated assistance with model formulation and with verification that a design for an electro-mechanical device satisfies its functional specification.

Introduction

The Stanford How Things Work Project is an ongoing integrated collection of research activities in the Knowledge Systems Laboratory at Stanford University [Fikes, et al 91] led by Richard Fikes. The overall objective of the project is to develop knowledge-based technology that will enable computer systems to offer intellectual assistance at high levels of competence to problem solvers and decision makers in all stages of the life cycle of engineered products. To achieve that objective, we are developing:

- Technology for representing knowledge about engineered devices in a form that enables the knowledge to be used in multiple systems for multiple reasoning tasks, and
- Reasoning methods that enable the represented knowledge to be effectively applied to the performance of the core engineering task of simulating and analyzing device behavior.

The knowledge to be represented includes a broad range of subjects, from the fundamentals of physics and engineering, to device models that describe device structure, behavior, and function, to the rationale for the design of specific devices. In order to directly support the reuse of encoded engineering knowledge bases, we are working with other research groups to establish a common device modeling language and a clearing house for device models. The common language will be based on and make use of the languages and tools being developed in the DARPA Knowledge-Sharing Initiative [Neches, et al 91].

---

1 This research sponsored by DARPA and NASA under NASA grants NAG 2-581 and NCC 2-537.

The primary engineering task on which we are focusing is that of supporting the design of electromechanical devices by providing effective tools for *simulating and analyzing the behavior of such devices* in all stages of their design. Simulation technology has the potential of providing a rapid low-cost means of testing new designs for sophisticated equipment in many industries before acquisition decisions are made and expensive prototypes are built. In order to realize that potential, simulators need to have three key capabilities they are currently lacking. Namely, simulators need to be:

- **Applicable to partially specified designs** -- Many of the financially significant decisions about new designs are made during conceptual and preliminary design stages. *Qualitative* simulation techniques are needed to obtain behavior analyses during those early stages of design, since the detailed design specifics required to do conventional numerical simulations are not yet available.
- **Rapidly reconfigurable** -- Simulators need to be capable of supporting a broad range of tests of a new design that include variations in level of detail (from engineering analyses of individual subsystems to macro level mission effectiveness evaluations of the overall design), issues being addressed (fuel consumption rates, ease of operation, response speed, etc.), and operating conditions being considered (extreme weather, variations in operator training, etc.). No one simulator or one simulation model will be able to support such a range of tests. Thus, designers need to be provided with a *simulation foundry* that enables them to rapidly configure and run multiple simulations on an as-needed basis to answer specific analysis questions.
- **Self interpreting** -- In order for designers to use simulators for multiple purposes on a routine basis, simulation results must be understandable with minimum effort. Thus, simulators need to provide significant assistance with the task of interpreting their output by producing summaries, explanations, and analyses which are directly oriented to a given analysis task.

We are developing knowledge-based technology that will remove those deficiencies. That is, we are developing augmentations to conventional simulators that will enable them to become applicable to partially specified designs, rapidly reconfigurable, and self-interpreting. In particular, we are developing techniques for:

- Automatically formulating a simulation model that embodies the abstractions, approximations, assumptions, and perspectives that are appropriate for a given analysis task,
- Performing qualitative simulation of device modules which have not yet been designed in detail or whose detailed quantitative behavior is not relevant to a given analysis task,
- Automatically guiding a simulator to consider scenarios that are relevant to a given analysis task,
- Generating human-understandable *causal* explanations of simulation results,
- Automatically determining whether simulated behavior satisfies functional specifications, and
- Testing and automatically generating procedures for operating the device.

We are embodying the techniques developed in our research in an evolving prototype "designer's associate" system called the *Device Modeling Environment (DME)* [Iwasaki and Low 91]. The DME system is intended to be useful to the research community at large as an experimental testbed, educational tool, and foundation on which to build new representation and reasoning capabilities. DME has already been developed to a

sufficient level of maturity to provide both a demonstration vehicle and a useful experimental testbed within the project.

DME is intended to enable a designer to document a design as it evolves and to support experimentation with alternative designs. The current system is used as follows:

- **Designer describes device** -- The designer selects components from a library and specifies the structural connections among the components.
- **Designer selects behavior models** -- The designer selects from a library the models of component behavior that provide the abstractions, approximations, assumptions, and perspectives which are appropriate for the analysis he or she wants to do.
- **DME generates simulation model** -- DME uses the device model specified by the designer to generate a qualitative *or* quantitative simulation model of the device .
- **Designer interactively guides the simulation** -- The designer uses a simulator provided by DME to interactively explore possible device behaviors .
- **DME provides causal explanations of simulated behavior**.
- **Designer analyzes behavior** -- The designer compares the predicted behavior with the intended functionality of the device.

New Capabilities Being Produced

The current DME system embodies state of the art research results. It provides an integrated set of tools for performing what might be characterized as a limited form of semi-automatic behavior analysis. For example, the system automatically formulates a simulation model, but only after the designer has selected from the system's model library appropriate behavior models for each device component.

Our current research is focused on taking steps toward providing a designer with a *comprehensive* and *fully automated* behavior analysis of a device being designed. Our three year goal is to develop new capabilities and integrate them into DME so that the system could be used as follows:

- **Designer describes device** -- In addition to the current facilities for selecting components from a library and specifying the structural connections among the components, new facilities will be developed to enable the designer to specify:
  - Intended functionality of a device,
  - Expected operator interactions with a device,
  - Assumptions about the environment in which a device will be operating, and
  - Rationale for design decisions;
- **DME formulates appropriate behavior model** -- New facilities will be developed that will enable DME to determine the abstractions, approximations, assumptions, and perspectives that are appropriate for specific analysis tasks such as testing whether the device design satisfies the functional specifications.
- **DME generates appropriate simulation model** -- DME will use the structural and behavioral device models to generate a simulation model of the device that *intermixes* qualitative and quantitative simulation as needed. New facilities will be developed to enable it to select an appropriate qualitative or quantitative simulator for each device

module depending on the level of detail at which the module has been designed and the level of detail required by the analysis task.

- **DME guides the simulation** -- New facilities will be developed to enable DME to direct the simulator to consider scenarios that are relevant to a given analysis goal such as testing whether the functional specification is satisfied.
- **DME determines whether behavior achieves the intended functionality** – New facilities will be developed to enable DME to compare the simulated behavior with the functional specification. In cases where the behavior does not satisfy the specifications, DME will be able to provide feedback in the form of additional constraints on the design which would guarantee that the device behaves as intended.
- **DME explains behavior analysis results** -- In addition to the current facilities for providing causal explanations of simulated behavior, new facilities will be developed to explain how and why the design either does or does not satisfy the functional specification.

An additional significant capability being developed in our project which is not highlighted in the above scenario is the use of DME for designing and analyzing procedures for operating a device. For example, DME seems particularly useful for assisting with the verification of procedures that respond to device malfunctions in that it enables simulation models to be rapidly reformulated to reflect malfunctions and can explain the effects caused by the procedures. We are currently working with NASA on such a procedure verification application in which DME will be used for both procedure debugging and operator training.

The central new capabilities currently being developed for DME are automated assistance with model formulation, automated assistance with verification that a design satisfies its functional specification, and automatic generation of causal explanations of device behavior. Our approach to achieving these capabilities is summarized in the sections below.

Automatic Model Formulation

We are developing methods for providing automated assistance with the core problem of model formulation -- a service that will help engineers build nontrivial models of device behavior for specific purposes.

The state of the art in model formulation today is model configuration from libraries of component models. Simulators such as SPICE [Katzenelson 66] and those for VHDL [Harr and Stanculescu 89] are based on libraries of component models which have been preformulated by modeling experts. The user selects components and configures them, and then the system compiles the code necessary to run a simulation and plot the trajectories of variables.

Today's component-based model libraries are most successful in those domains where components are well-defined idealizations at a single level of abstraction, such as logical circuits. The mapping between physical components and idealized component models is simple, and there is exactly one behavior model associated with each component model. Thus, the engineer's part of the model formulation task is simplified in that he or she need only specify a component connection topology.

However, in most domains and tasks, the mapping from phenomena of interest in a physical system to a set of possible behavior models is complex and the result of nontrivial

reasoning. In doing model formulation, an engineer must identify the *relevant abstractions* to model, deciding, for example, whether to treat the load of an electrical power supply as a single resistive element or as a system with components that vary in their power usage. The engineer must also make *simplifying assumptions* and *approximations*, such as to assume no friction in a gear train or that a valve can be modeled as a discrete switch. The engineer makes these modeling choices to produce a model which answers a particular information need in a reasonable amount of time.

The power of the library approach derives from the reuse of the component models and the automatic assembly of system models from partial descriptions. DME will achieve the same advantages of knowledge reuse and automation, but for a more general class of domains and for multiple modeling purposes.

Even in domains such as analog circuits where there is a large library of ready-made simulation modules for standard components, building a model of an entire system is not an easy task. There are typically many simulation modules for each type of component, each based on different simplifying assumptions and approximations which are not stated explicitly. Therefore, a significant amount of effort and expertise is required for engineers to use even off-the-shelf simulation modules to assemble a model of a whole system. Engineers often prefer to write their own modules instead of using off-the-shelf modules precisely because they do not know all the underlying assumptions and do not trust their results.

DME will enable *knowledge reuse* by providing the representation and architecture for model libraries containing a comprehensive body of behavior model fragments, each making particular abstractions and approximations and conditioned on *explicitly represented* modeling assumptions. The formalism and examples will allow engineers to fill the libraries with model fragments covering those phenomena they need to model. We expect that a market will develop for these models, possibly driving a small industry of component-model-building specialists (as in electronics).

DME will provide *automated model formulation assistance* using these libraries. The assistance will change the nature of the interaction between the human engineer and the computational environment. Instead of operating at the level of equations or fixed-level component models, the engineer may specify the high-level device structure, the simulation goal, the utility criteria, a description of the context of use, and any initial conditions. The system will take an active role in selecting appropriate model fragments to construct a complete and coherent simulation model. This advance in model formulation is similar to the improvement in software development from early assembly-language programming to Fourth Generation Language environments.

Automatic Behavior Verification

Understanding the design of an engineered device requires both knowledge of the general physical principles that determine the behavior of the device and knowledge of what the device is intended to do (i.e., its functional specification). However, the majority of work in model-based reasoning about device behavior has focused on modeling a device in terms of general physical principles *or* intended functionality, but not both. For example, most of the work in qualitative physics has been concerned with predicting the behavior of a device given its physical structure and knowledge of general physical principles. In that work, great importance has been placed on preventing a pre-conceived notion of an intended function of the device from influencing the system's reasoning methods and representation of physical principles in order to guarantee a high level of "objective truth"

in the predicted behavior. In contrast, in their work based on the FR (Functional Representation) language [Sembugamoorthy & Chandrasekaran, 1986] [Keuneke, 1991], Chandrasekaran and his colleagues have focused mostly on modeling a device in terms of what the device is intended to do and how those intentions are to be accomplished through causal interactions among components of the device.

Both types of knowledge, functional and behavioral, seem to be indispensable in fully understanding a device design. On the one hand, knowledge of intended function alone does not enable one to reason about what a device might do when it is placed in an unexpected condition or to infer the behavior of an unfamiliar device from its structure. On the other hand, knowledge of device structure and general physical principles may allow one to predict how the device will behave under a given condition, but without knowledge of the intended functions, it is impossible to determine if the predicted behavior is a desirable one, or what aspect of the behavior is significant.

In order to use both functional and behavioral knowledge in understanding a device design, it is crucial that the functional knowledge is represented in such a way that it has a clear interpretation in terms of actual behavior. Suppose, for example, that the function of a charge current controller is to prevent damage to a battery by cutting off the charge current when the battery is fully charged. To be able to determine whether this function is actually accomplished by an observed behavior of the device, the representation of the function must specify conditions that can be evaluated against the behavior. Such conditions might include occurrence of a temporal sequence of expected events and causal relations among the events and the components. Without a clear semantics given to a representation of functions in terms of actual behavior, it would be impossible to evaluate a design based on its predicted behavior and intended functions.

While it is important for a functional specification to have a clear interpretation in terms of actual behavior, it is also desirable for the language for specifying functions to be independent of any particular system used for simulation. Though there are a number of alternative methods for predicting behavior, such as numerical simulation with discrete time steps or qualitative simulation, a functional specification at some abstract level should be intuitively understandable without specifying a particular simulation mechanism. If a functional specification language was dependent on a specific simulation language or mechanism, a separate functional specification language would be needed for each different simulation language, which is clearly undesirable. What is needed is a functional specification language that has sufficient expressive power to support descriptions of the desired functions of a variety of devices. At the same time, the language should be clear enough so that for each simulation mechanism used, it can be given an unambiguous interpretation in terms of a simulated behavior.

An essential element in the description of a function is causality. In order to say that a device has achieved a function, which may be expressed as a condition on the state of the world, one must show not only that the condition is satisfied but also that the device has participated in the causal process that has brought about the condition. For example, when an engineer designs a thermostat to keep room temperature constant, the design embodies her idea about how the device is to work. In fact, the essential part of her knowledge of its function is the expected causal chain of events in which it will take part in achieving the goal. Thus, a representation formalism of functions must provide a means of expressing knowledge about such causal processes.

We are developing a new representational formalism for specifying device functions called CFRL (Causal Functional Representation Language) that allows functions to be expressed in terms of expected causal chains of events [Vescovi, Iwasaki, Fikes, &

Chandrasekaran, 1993]. We are providing the language with a well-defined semantics in terms of the type of behavior representation widely used in model-based, qualitative simulation. Finally, we are using CFRL as the basis for a function verification program which determines whether a behavior achieves an intended function.

Explanation Generation

We are developing a method for generating explanations of how devices work and incorporating that method in DME [Gruber & Gautier, 1992; Gruber & Gautier, 1993]. On the basis of an initial device model and the behavioral predictions obtained through simulation, DME can answer a range of user queries about the structure and behavior of the modeled system.

The approach we are developing combines several techniques for explanation generation:

- Automatically synthesizing formal mathematical models from high-level model specifications, and explaining low-level simulation data in terms of the original specifications
- Inferring causality from mathematical models, rather than assuming ad hoc, hand-crafted causal models
- Dynamically generating explanations in response to user queries, which are formulated by direct manipulation on text and graphics displayed during simulation
- Supporting interactive follow-up questions, allowing the user to get more information on a particular point of an explanation
- Using a compositional method of text generation, in which textual annotations of model fragments are composed into phrases, which are then processed to produce smooth, concise text.

The explanations are intended for three application tasks: data interpretation, the design of operator procedures, and design documentation. The task of interpreting simulation data is important for exploring hypotheses about device behavior during conceptual design and for debugging the simulation model itself. Machine-generated explanations can facilitate data interpretation by showing the relationship between low-level simulation data and the original modeling decisions and assumptions. In the second application, operators of equipment need to rapidly explore failure scenarios in order to design and test corrective procedures. Dynamically generated causal explanations can help the operator assess the situation and determine appropriate actions. Finally, self-explanatory simulations can be used to document design intent [Gruber, 1990; Gruber, 1991]. Instead of writing a static design document that is often inaccurate and out of date, the designer can demonstrate the intended and expected behavior of a device using simulation. The system can generate explanations in response to questions by the reader.

An important element of the explanation approach in DME is the use of real engineering models, rather than ad hoc "causal models" that are built specifically for explanation generation. In explaining how things work, people *do* use causal terminology. However, when analyzing the behavior of devices, engineers use formalisms such as logical and mathematical constraints that are not causal. DME *infers causal dependencies* among modeled parameters by analyzing logical and mathematical constraints.

In DME, logical constraints occur in the preconditions of model fragments. Discrete events, such as changes in the operating regions of components or discontinuous changes in quantitative parameters, are due to changes in the activation of model fragments. The

"cause" of a discrete event, then, can be viewed as the set of facts and parameter values that satisfied the preconditions of a model fragment representing the event. DME can therefore explain the cause of discrete events by describing how the preconditions of model fragments are satisfied. It can then recursively explain the causal ancestry of each of the facts or variable values in the preconditions. This is similar to the traditional approach to explaining rule firings in expert systems. In DME, heuristics are applied to filter some of the facts and variables, producing a more concise explanation.

The collection of techniques we are developing constitute a practical method for generating interactive explanations of device behavior in natural language. No special knowledge of linguistics is needed for building models; the engineer merely annotates behavior models developed for simulation. Because causal relationships are inferred for each simulation scenario, there is no need to build in assumptions of causality in the models. The modeling and simulation technology that underlies the approach is realistic for physical systems that can be modeled with time-varying ordinary differential equations, such as electromechanical devices for controlling position or force (e.g., robot manipulators), and process control systems (e.g., control of fuel supply for the Space Shuttle).

Bibliography

[Cutkosky et al 93] Cutkosky, M. R., Engelmore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, T. R., Mark, W. S., Tenenbaum, J. M., & Weber, J. C. (1993). PACT: An Experiment in Integrating Concurrent Engineering Systems. IEEE Computer, 26(1), pp. 28-37. Also available as KSL 93-21.

[Fikes, et al 91] Richard Fikes, Tom Gruber, Yumi Iwasaki, Alon Levy, and Pandu Nayak; "How Things Work Project Overview"; Knowledge Systems Laboratory Technical Report KSL 91-70; Computer Science Department, Stanford University; November 1991.

[Gautier & Gruber 93] Gautier, P. O. & Gruber, T. R. (1993). Generating Explanations of Device Behavior Using Compositional Modeling and Causal Ordering. Eleventh National Conference on Artificial Intelligence. July 1993. AAAI Press. Also available as KSL 93-06.

[Gruber 92] Gruber, T. (1992). Model Formulation as a Problem Solving Task: Computer-assisted Engineering Modeling. International Journal of Intelligent Systems, 8(1) , 105-127. KSL 92-57.

[Gruber & Gautier 93] Gruber, T. & Gautier, P. (1992). Machine-generated Explanations of Engineering Models: A compositional modeling approach. To appear in the Proceedings of the 1993 International Joint Conference on Artificial Intelligence. August 1993. Also available as KSL 92-85.

[Gruber and Russell 92] Gruber, T. & Russell, D. M. (1992). Derivation and Use of Design Rationale Information as Expressed by Designers. Also available as KSL 92-64.

[Gruber and Russell 92a] Gruber, T. & Russell, D. M. (1992). Generative Design Rationale: Beyond the Record and Replay Paradigm. Collection on Design Rationale. Kluwer Academic Publishers. Also available as KSL 92-59.

[Gruber 92a] Gruber, T. R. (1992). A Translation Approach to Portable Ontology Specifications. Second Japanese Knowledge Acquisition Workshop, Kobe and Tokyo, Japan. Also available as KSL 92-71.

[Gruber 93] Gruber, T. R. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. KSL 93-04.

[Harr and Stanculescu 89] R. Harr and A. Stanculescu (Eds.); *Applications of VHDL to Circuit Design*; Kluwer Academic Publishers; Norwell, Massachusetts; 1989.

[Iwasaki et al 93] Iwasaki, Y., Fikes, R., Vescovi, M., & Chandrasekaran, B. (1993). How Things are Intended to Work: Capturing Functional Knowledge in Device Design. To appear in the Proceedings of the 1993 International Joint Conference on Artificial Intelligence. August 1993. Also available as KSL 93-39.

[Iwasaki & Levy 93] Iwasaki, Y. & Levy, A. Y. (1993). Automated Model Selection for Simulation. QR93. Also available as KSL 93-11.

[Iwasaki and Chandrasekaran 92] Y. Iwasaki and B. Chandrasekaran; "Design Verification Through Function and Behavior-Oriented Representations: Bridging the Gap Between Function and Behavior"; Proceedings of the Second International Conference On Artificial Intelligence in Design; 1992.

[Iwasaki and Low 91] Y. Iwasaki and C. M. Low; "Model Generation and Simulation of Device Behavior with Continuous and Discrete Changes: the Details"; Technical Report, KSL 91-30, Knowledge Systems Laboratory, Stanford University, 1991.

[Katzenelson 66] J. Katzenelson; "AEDNET: A simulator for nonlinear networks"; Proceedings of the IEEE, 54(11).

[Levy, et al 92] Levy, A. Y., Iwasaki, Y., & Motoda, H. (1992). Relevance Reasoning to Guide Compositional Modeling. Second Pacific Rim International Conference on Artificial Intelligence, Seoul, Korea. Also available as KSL 92-47.

[Levy et al 93] Levy, A. Y., Mumick, I. S., Sagiv, Y., & Shmueli, O. (1993). Equivalence, Query-reachability and Satisfiability in Datalog Extensions. Twelfth ACM-SIGACT-SIGART-SIGMOD Symposium on Principles of Database System (PODS-93). Also available as KSL 93-15.

[Levy & Sagiv 93] Levy, A. Y. & Sagiv, Y. (1993). Controlling Inference Using the Query Tree. Biannual Israeli Symposium on Foundations of AI-93. Also available as KSL 93-07.

[Levy & Sagiv 93a] Levy, A. Y. & Sagiv, Y. (1993). Exploiting Irrelevance-reasoning to Guide Problem Solving. To appear in the Proceedings of the 1993 International Joint Conference on Artificial Intelligence. August 1993. Also available as KSL 93-05.

[Levy & Sagiv 93b] Levy, A. Y. & Sagiv, Y. (1993). Queries Independent of Updates. Twelfth ACM-SIGACT-SIGART-SIGMOD Symposium on Principles of Database System (PODS-93). Also available as KSL 93-03.

[Nayak 92] Nayak, P.P. Automated Modeling of Physical Systems. (Ph.D. Dissertation) September 1992.

[Nayak 92a] Nayak, P. P. Order of Magnitude Reasoning Using Logarithms. Third International Conference on the Principles of Knowledge Representation and Reasoning. October 1992. Morgan Kaufman. Also available as KSL 92-29.

[Nayak 92b] Nayak, P. P. Causal Approximations. Tenth National Conference on Artificial Intelligence. July 1992. AAAI Press. Pgs. 703-709.

[Nayak et al 92] Nayak, P. P., Joskowicz, L., Addanki, S. Automated Model Selection using Context-Dependent Behaviors. Tenth National Conference on Artificial Intelligence. July 1992. AAAI Press. Pgs. 710-716.

[Neches, et al 91] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, & W. Swartout; "Enabling Technology for Knowledge Sharing"; *AI Magazine*, vol. 12, no. 3, pp 16-36; 1991.

[Patil et al 92] Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., Mckay, D., Finin, T., Gruber, T. R., & Neches, R. (1992). The DARPA Knowledge Sharing Effort: Progress Report. Principles of Knowledge Representation and Reasoning, Cambridge, MA, 777-788, Morgan Kaufmann. Also available as KSL 93-23.

[Sembugamoorthy & Chandrasekaran 86] V. Sembugamoorthy and B. Chandrasekaran; "Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems"; in J. L. Kolodner and C. K. Riesbeck (editors); *Experience, Memory, and Reasoning*; Lawrence Erlbaum Associates, Hillsdale, NJ; 1986.

[Vescovi, et al 93] Vescovi, M., Iwasaki, Y., Fikes, R., & Chandrasekaran, B. (1993). CFRL: A Language for Specifying the Causal Functionality of Engineering Devices. Eleventh National Conference on Artificial Intelligence. July 1993. AAAI Press. Also available as KSL 93-38.

N94- 34042

Acting to Gain Information: Real-time Reasoning Meets Real-time Perception

Stan Rosenschein
Teleos Research

Recent advances in intelligent reactive systems suggest new approaches to the problem of deriving task-relevant information from perceptual systems in real time. The author will describe work in progress aimed at coupling intelligent control mechanisms to real-time perception systems, which special emphasis on frame rate visual measurement systems. A model for integrated reasoning and perception will also be discussed, the special challenges associated with visual information processing will be discussed, and recent progress in applying these ideas to problems of sensor utilization for efficient recognition and tracking will be described.

I. roceedings ... he Ninth Conference on Uncertainty in Artificial Intelligence, Washington, D.C., July 1993.

TECHNICAL REPORT
R-192-II
May 1993

N94-34043

From Conditional Oughts to Qualitative Decision Theory

Judea Pearl
Cognitive Systems Laboratory
University of California, Los Angeles, CA 90024
judea@cs.ucla.edu
Content Areas: Commonsense Reasoning, Probabilistic Reasoning, Reasoning about Action

Abstract

The primary theme of this investigation is a decision theoretic account of conditional ought statements (e.g., "You ought to do A, if C") that rectifies glaring deficiencies in classical deontic logic. The resulting account forms a sound basis for *qualitative decision theory*, thus providing a framework for qualitative planning under uncertainty. In particular, we show that adding causal relationships (in the form of a single graph) as part of an epistemic state is sufficient to facilitate the analysis of action sequences, their consequences, their interaction with observations, their expected utilities and, hence, the synthesis of plans and strategies under uncertainty.

1 INTRODUCTION

In natural discourse, "ought" statements reflect two kinds of considerations: requirements to act in accordance with moral convictions or peer's expectations, and requirements to act in the interest of one's survival, namely, to avoid danger and pursue safety. Statements of the second variety are natural candidates for decision theoretic analysis, albeit qualitative in nature, and these will be the focus of our discussion. The idea is simple. A sentence of the form "You ought to do A if C" is interpreted as shorthand for a more elaborate sentence: "If you observe, believe, or know C, then the expected utility resulting from doing A is much higher than that resulting from not doing A".[1] The longer sentence combines several modalities that have been the subjects of AI investigations: observation, belief, knowledge, probability ("expected"), desirability ("utility"), causation ("resulting from"), and, of course, action ("doing A"). With the exception of utility, these modalities have been formulated recently using qualitative, order-of-magnitude abstractions of probability theory (Goldszmidt & Pearl 1992, Goldszmidt 1992). Utility preferences themselves, we know from decision theory, can be fairly unstructured, save for obeying asymmetry and transitivity. Thus, paralleling the order-of-magnitude abstraction of probabilities, it is reasonable to score consequences on an integer scale of utility: very desirable ($U = O(1/\epsilon)$), very undesirable ($U = -O(1/\epsilon)$), bearable ($U = O(1)$), and so on, mapping each linguistic assessment into the appropriate $\pm\, O(1/\epsilon^i)$ utility rating. This utility rating, when combined with the infinitesimal rating of probabilistic beliefs (Goldszmidt & Pearl 1992), should permit us to rate actions by the expected utility of their consequences, and a requirement to do A would then be asserted iff the rating of doing A is substantially (i.e., a factor of $1/\epsilon$) higher than that of not doing A.

This decision theoretic agenda, although conceptually straightforward, encounters some subtle difficulties in practice. First, when we deal with actions and consequences, we must resort to causal knowledge of the domain and we must decide how such knowledge is to be encoded, organized, and utilized. Second, while theories of actions are normally formulated as theories of temporal changes (Shoham 1988, Dean & Kanazawa 1989), ought statements invariably suppress explicit references to time, strongly suggesting that temporal information is redundant, namely, it can be reconstructed if required, but glossed over otherwise. In other words, the fact that people comprehend, evaluate and follow non-temporal ought statements suggests that people adhere to some canonical, yet implicit assumptions about temporal progression of events, and that no account can be complete without making these assumptions explicit. Third, actions in decision theory are predesignated explicitly to a few distinguished atomic variables, while statements of the type "You ought to do A" are presumed applicable to any arbitrary proposition A.[2] Finally, decision theoretic methods, especially those based on static influence diagrams, treat both the informational relationships between observations and actions and the causal relationships between actions and consequences as instantaneous (Chapter 6, Pearl 1988, Shachter 1986). In real-

[1] An alternative interpretation, in which "doing A" is required to be substantially superior to both "not doing A" and "doing not-A" is equally valid, and could be formulated as a straightforward extension of our analysis.

[2] This has been an overriding assumption in both the deontic logic and the preference logic literatures.

ity, the effect of our next action might be to invalidate currently observed properties, hence any non-temporal account of ought must carefully distinguish properties that are influenced by the action from those that will persist despite the action, and must explicate therefore some canonical assumptions about persistence.

These issues are the primary focus of this paper. We start by presenting a brief introduction to infinitesimal probabilities and showing how actions, beliefs, and causal relationships are represented by ranking functions $\kappa(\omega)$ and causal networks Γ (Section 2). In Section 3 we present a summary of the formal results obtained in this paper, including an assertability criterion for conditional oughts. Sections 4 and 5 explicate the assumptions leading to the criterion presented in Section 3. In Section 4 we introduce an integer-valued utility ranking $\mu(\omega)$ and show how the three components, $\kappa(\omega)$, Γ, and $\mu(\omega)$, permit us to calculate, semi-qualitatively, the utility of an arbitrary proposition φ, the utility of a given action A, and whether we ought to do A. Section 5 introduces conditional oughts, namely, statements in which the action is contingent upon observing a condition C. A calculus is then developed for transforming the conditional ranking $\kappa(\omega|C)$ into a new ranking $\kappa_A(\omega|C)$, representing the beliefs an agent will possess after implementing action A, having observed C. These two ranking functions are then combined with $\mu(\omega)$ to form an assertability criterion for the conditional statement $O(A|C)$: "We ought to do A, given C". In Section 6 we compare our formulation to other accounts of ought statements, in particular deontic logic, preference logic, counterfactual conditionals, and quantitative decision theory.

2 INFINITESIMAL PROBABILITIES, RANKING FUNCTIONS, CAUSAL NETWORKS, AND ACTIONS

1. (*Ranking Functions*). Let Ω be a set of worlds, each world $\omega \in \Omega$ being a truth-value assignment to a finite set of atomic variables $(X_1, X_2, ..., X_n)$ which in this paper we assume to be bi-valued, namely, $X_i \in \{true, false\}$. A belief *ranking function* $\kappa(\omega)$ is an assignment of non-negative integers to the elements of Ω such that $\kappa(\omega) = 0$ for at least one $\omega \in \Omega$. Intuitively, $\kappa(\omega)$ represents the degree of surprise associated with finding a world ω realized, and worlds assigned $\kappa = 0$ are considered serious possibilities. $\kappa(\omega)$ can be considered an order-of-magnitude approximation of a probability function $P(\omega)$ by writing $P(\omega)$ as a polynomial of some small quantity ϵ and taking the most significant term of that polynomial, i.e.,

$$P(\omega) \cong C\epsilon^{\kappa(\omega)} \tag{1}$$

Treating ϵ as an infinitesimal quantity induces a conditional ranking function $\kappa(\varphi|\psi)$ on propositions which is governed by Spohn's calculus (Spohn 1988):

$$\kappa(\Omega) = 0$$

$$\kappa(\varphi) = \begin{cases} \min_\omega \kappa(\omega) & \text{for } \omega \models \varphi \\ \infty & \text{for } \omega \models \neg\varphi \end{cases}$$

$$\kappa(\varphi|\psi) = \kappa(\varphi \wedge \psi) - \kappa(\psi) \tag{2}$$

2. (*Stratified Rankings and Causal Networks* (Goldszmidt & Pearl 1992)). A *causal network* is a directed acyclic graph (dag) in which each node corresponds to an atomic variable and each edge $X_i \longrightarrow X_j$ asserts that X_i has a direct causal influence on X_j. Such networks provide a convenient data structure for encoding two types of information: how the initial ranking function $\kappa(\omega)$ is formed, and how external actions would influence the agent's belief ranking $\kappa(\omega)$. Formally, causal networks are defined in terms of two notions: stratification and actions.

A ranking function $\kappa(\omega)$ is said to be *stratified* relative to a dag Γ if

$$\kappa(\omega) = \sum_i \kappa(X_i(\omega)|\mathbf{pa}_i(\omega)) \tag{3}$$

where $\mathbf{pa}_i(\omega)$ are the parents of X_i in Γ evaluated at state ω. Given a ranking function $\kappa(\omega)$, any edge-minimal dag Γ satisfying Eq. (3), is called a *Bayesian network* of $\kappa(\omega)$ (Pearl 1988). A dag Γ is said to be a causal network of $\kappa(\omega)$ if it is a Bayesian network of $\kappa(\omega)$ and, in addition, it admits the following representation of actions.

3. (*Actions*) The effect of an atomic action $do(X_i = true)$ is represented by adding to Γ a link $DO_i \longrightarrow X_i$, where DO_i is a new variable taking values in $\{do(x_i), do(\neg x_i), idle\}$ and x_i stands for $X_i = true$. Thus, the new parent set of X_i is $\mathbf{pa}'_i = \mathbf{pa}_i \cup \{DO_i\}$ and it is related to X_i by

$$\kappa(X_i(\omega)|\mathbf{pa}'_i(\omega)) = \begin{cases} \kappa(X_i(\omega)|\mathbf{pa}_i(\omega)) & \text{if } DO_i = idle \\ \infty & \text{if } DO_i = do(y) \text{ and } X_i(\omega) \neq y \\ 0 & \text{if } DO_i = do(y) \text{ and } X_i(\omega) = y \end{cases} \tag{4}$$

The effect of performing action $do(x_i)$ is to transform $\kappa(\omega)$ into a new belief ranking, $\kappa_{x_i}(\omega)$, given by

$$\kappa_{x_i}(\omega) = \begin{cases} \kappa'(\omega|do(x_i)) & \text{for } \omega \models x_i \\ \infty & \text{for } \omega \models \neg x_i \end{cases} \tag{5}$$

where κ' is the ranking dictated by the augmented network $\Gamma \cup \{DO_i \longrightarrow X_i\}$ and Eqs. (3) and (4).

This representation embodies the following aspects of actions:

(i) An action $do(x_i)$ can affect only the descendants of X_i in Γ.

(ii) Fixing the value of $\mathbf{pa}_i$ (by some appropriate choice of actions) renders X_i unaffected by any external intervention $do(x_\kappa), \kappa \neq i$.

3 SUMMARY OF RESULTS

The assertability condition we are about to develop in this paper requires the specification of an epistemic state $ES = (\kappa(\omega),\ \Gamma,\ \mu(\omega))$ which consists of three components:

$\kappa(\omega)$ - an ordinal belief ranking function on Ω.

Γ - a causal network of $\kappa(\omega)$.

$\mu(\omega)$ - an integer-valued utility ranking of worlds, where $\mu(\omega) =\pm\ i$ assigns to ω a utility $U(\omega) =\pm\ O(1/\epsilon^i), i = 0, 1, 2, ...$

The main results of this paper can be summarized as follows:

1. Let W_i^+ and W_i^- be the formulas whose models receive utility ranking $+i$ and $-i$, respectively, and let $\kappa'(\omega)$ denote the ranking function that prevails after establishing the truth of event φ, where φ is an arbitrary proposition (i.e., $\kappa'(\neg\varphi) = \infty$ and $\kappa'(\varphi) = 0$). The expected utility rank of φ is characterized by two integers

$$\begin{aligned} n^+ &= \max_i[0;\ i - \kappa'(W_i^+ \wedge \varphi)] \\ n^- &= \max_i[0;\ i - \kappa'(W_i^- \wedge \varphi)] \end{aligned} \tag{6}$$

and is given by

$$\mu[(\varphi;\ \kappa'(\omega)] = \begin{cases} \text{ambiguous} & \text{if } n^+ = n^- > 0 \\ n^+ - n^- & \text{otherwise} \end{cases} \tag{7}$$

2. A conditional ought statement $O(A|C)$ is assertable in ES iff

$$\mu(A; \kappa_A(\omega|C)) > \mu(true; \kappa(\omega|C)) \tag{8}$$

where A and C are arbitrary propositions and the ranking $\kappa_A(\omega|C)$ (to be defined in step 3) represents the beliefs that an agent anticipates holding, after implementing action A, having observed C.

3. If A is a conjunction of atomic propositions, $A = \bigwedge_{j\in J} A_j$, where each A_j stands for either $X_j = true$ or $X_j = false$, then the post-action ranking $\kappa_A(\omega|C)$ is given by the formula

$$\kappa_A(\omega|C) = \kappa(\omega) - \sum_{i\in I\cup R} \kappa(X_i(\omega)|\mathbf{pa}_i(\omega)) + \min_{\omega'}[\sum_{i\in J} S_i(\omega,\omega') + \kappa(\omega'|C)] \tag{9}$$

where R is the set of root nodes and

$$S_i(\omega,\omega') = \begin{cases} s_i & \text{if } X_i(\omega) \neq X_i(\omega') \text{ and } \mathbf{pa}_i = \emptyset \\ s_i & \text{if } X_i(\omega) \neq X_i(\omega'), \mathbf{pa}_i \neq \emptyset \text{ and} \\ & \quad \kappa(\neg X_i(\omega)|\mathbf{pa}_i(\omega)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$S(\omega,\omega')$ represents persistence assumptions: It is surprising (to degree $s_i \geq 1$) to find X_i change from its pre-action value of $X_i(\omega')$ to a post-action value of $X_i(\omega)$ if there is no causal reason for the change.

If A is a disjunction of actions, $A = \bigvee_l A^l$, where each A^l is a conjunction of atomic propositions, then

$$\kappa_A(\omega|C) = \min_l \kappa_{A^l}(\omega|C) \tag{11}$$

4 FROM UTILITIES AND BELIEFS TO GOALS AND ACTIONS

Given a proposition φ that describes some condition or event in the world, what information is needed before we can evaluate the merit of obtaining φ, or, at the least, whether φ_1 is "preferred" to φ_2? Clearly, if we are to apply the expected utility criterion, we should define two measures on possible worlds, a probability measure $P(\omega)$ and a utility measure $U(\omega)$. The first rates the likelihood that a world ω will be realized, while the second measures the desirability of ω. Unfortunately, probabilities and utilities in themselves are not sufficient for determining preferences among propositions. The merit of obtaining φ depends on at least two other factors: how the truth of φ is established, and what control we possess over which model of φ will eventually prevail. We will demonstrate these two factors by example.

Consider the proposition φ = "The ground is wet". In the midst of a drought, the consequences of this statement would depend critically on whether we watered the ground (action) or we happened to find the ground wet (observation). Thus, the utility of a proposition φ clearly depends on how we came to know φ, by mere observation or by willful action. In the first case, finding φ true may provide information about the natural process that led to the observation φ, and we should change the current probability from $P(\omega)$ to $P(\omega|\varphi)$. In the second case, our actions may perturb the natural flow of events, and $P(\omega)$ will change without shedding light on the typical causes of φ. We will denote the probability resulting from externally enforcing the truth of φ by $P_\varphi(\omega)$, which will be further explicated in Section 5 in terms of the causal network Γ.[3]

However, regardless of whether the probability function $P(\omega|\varphi)$ or $P_\varphi(\omega)$ results from learning φ, we are still unable to evaluate the merit of φ unless we understand what control we have over the opportunities offered by φ. Simply taking the expected utility $U(\varphi) = \Sigma_\omega[P(\omega|\varphi)U(\omega)]$ amounts to assuming that the agent is to remain totally passive until Nature selects a world ω with probability $P(\omega|\varphi)$, as in a game of chance. It ignores subsequent actions which the agent might be able to take so as to change this probability. For example, event φ might provide the agent with the option of conducting further tests so as to determine with greater certainty which world would eventually be realized. Likewise, in case φ stands for "Joe went to get his gun", our agent might possess the wisdom to protect itself by escaping in the next taxicab.

[3]The difference between $P(\omega|\varphi)$ and $P_\varphi(\omega)$ is precisely the difference between conditioning and "imaging" (Lewis 1973), and between belief revision and belief update (Alchourron et.al. 1985, Katsuno & Mendelzon 1991, Goldszmidt & Pearl 1992). It also accounts for the difference between indicative and subjunctive conditionals - a topic of much philosophical discussion (Harper et.al. 1980).

In practical decision analysis the utility of being in a situation φ is computed using a dynamic programming approach, which assumes that subsequent to realizing φ the agent will select the optimal sequence of actions from those enabled by φ. This computation is rather exhaustive and is often governed by some form of myopic approximation (Chapter 6, Pearl 1988). Ought statements normally refer to a single action A, tacitly assuming that the choice of subsequent actions, if available, is rather obvious and their consequences are well understood. We say, for example, "You ought to get some food", assuming that the food would subsequently be eaten and not be left to rot in the car. In our analysis, we will make a similar myopic approximation, assuming either that action A is terminal or that the consequences of subsequent actions (if available) are already embodied in the functions $P(\omega)$ and $\mu(\omega)$. We should keep in mind, however, that the result of this myopic approximation is not applicable to *all* actions; in sequential planning situations, some actions may be selected for the sole purpose of enabling certain subsequent actions.

Denote by $P'(\omega)$ the probability function that would prevail after obtaining φ.[4] Let us examine next how the expected utility criterion $U(\varphi) = \Sigma P'(\omega)U(\omega)$ translates into the language of ranking functions.

Let us assume that U takes on values in $\{-O(1/\epsilon), O(1), +O(1/\epsilon)\}$, read as {very undesirable, bearable, very desirable}. For notational simplicity, we can describe these linguistic labels as a *utility ranking* function $\mu(\omega)$ that takes on the values -1, 0, and $+1$, respectively. Our task, then, is to evaluate the rank $\mu(\varphi)$, as dictated by the expected value of $U(\omega)$ over the models of φ.

Let the sets of worlds assigned the ranks -1, 0, and $+1$ be represented by the formulas W^-, W^0, and W^+, respectively, and let the intersections of these sets with φ be represented by the formulas φ^-, φ^0, and φ^+, respectively. The expected utility of φ is given by $-C_-/\epsilon\ P'(W^-) + C_0\ P'(W^0) + C_+/\epsilon\ P'(W^+)$, where C_-, C_0, and C_+ are some positive coefficients. Introducing now the infinitesimal approximation for P', in the form of the ranking function κ', we obtain

$$U(\varphi) = \begin{cases} -O(1/\epsilon) & \text{if } \kappa'(\varphi^-) = 0 \text{ and } \kappa'(\varphi^+) > 0 \\ O(1) & \text{if } \kappa'(\varphi^-) > 0 \text{ and } \kappa'(\varphi^+) > 0 \\ +O(1/\epsilon) & \text{if } \kappa'(\varphi^-) > 0 \text{ and } \kappa'(\varphi^+) = 0 \\ \text{ambiguous} & \text{if } \kappa'(\varphi^-) = 0 \end{cases} \quad (12)$$

The ambiguous status reflects a state of conflict $U(\varphi) = -C_-/\epsilon + C_+/\epsilon$, where there is a serious possibility of ending in either terrible disaster or enormous success. Recognizing that ought statements are often intended to avert such situations (e.g., "You ought to invest in something safer"), we may take a risk-averse attitude and rank ambiguous states as low as $U = -O(1/\epsilon)$ (other attitudes are, of course, perfectly legitimate). This attitude, together with $\kappa'(\varphi) = 0$, yields the desired expression for $\mu(\varphi;\ \kappa'(\omega))$:

$$\mu(\varphi;\kappa'(\omega)) = \begin{cases} -1 & \text{if } \kappa'(W^-|\varphi) = 0 \\ 0 & \text{if } \kappa'(W^- \vee W^+|\varphi) > 0 \\ +1 & \text{if } \kappa'(W^-|\varphi) > 0 \text{ and } \kappa'(W^+|\varphi) = 0 \end{cases} \quad (13)$$

The three-level utility model is, of course, only a coarse rating of desirability. In a multi-level model, where W_i^+ and W_i^- are the formulas whose models receive utility ranking $+i$ and $-i$, respectively[5], and $i = 0, 1, 2, \ldots$, the ranking of the expected utility of φ is given by Eq. (7) (Section 3).

Having derived a formula for the utility rank of an arbitrary proposition φ, we are now in a position to formulate our interpretation of the deontic expression $O(A|C)$: "You ought to do A if C, iff the expected utility associated with doing A is much higher than that associated with not doing A". We start with a belief ranking $\kappa(\omega)$ and a utility ranking $\mu(\omega)$, and we wish to assess the utilities associated with doing A versus not doing A, given that we observe C. The observation C would transform our current $\kappa(\omega)$ into $\kappa(\omega|C)$. Doing A would further transform $\kappa(\omega|C)$ into $\kappa'(\omega) = \kappa_A(\omega|C)$, while not doing A would render $\kappa(\omega|C)$ unaltered, so $\kappa'(\omega) = \kappa(\omega|C)$. Thus, the utility rank associated with doing A is given by $\mu(A;\kappa'_A(\omega|C))$, while that associated with not doing A is given by $\mu(C;\kappa(\omega|C)) = \mu(true;\kappa(\omega|C))$. Consequently, we can write the assertability criterion for conditional ought as

$$O(A|C) \quad \text{iff} \quad \mu(A;\kappa_A(\omega|C)) > \mu(true;\kappa(\omega|C)) \quad (14)$$

where the function $\mu(\varphi;\kappa(\omega))$ is given in Eq. (13).

We remark that the transformation from $\kappa(\omega|C)$ to $\kappa_A(\omega|C)$ requires causal knowledge of the domain, which will be provided by the causal network Γ (Section 5). Once we are given Γ it will be convenient to encode both $\kappa(\omega)$ and $\mu(\omega)$ using integer-valued labels on the links of Γ. Moreover, it is straightforward to apply Eqs. (7) and (14) to the usual decision theoretic tasks of selecting an optimal action or an optimal information-gathering strategy (Chapter 6, Pearl 1988).

Example 1:
To demonstrate the use of Eq. (14), let us examine the assertability of "If it is cloudy you ought to take an umbrella" (Boutilier 1993). We assume three atomic propositions, c - "Cloudy", r - "Rain", and u - "Having an Umbrella", which form eight worlds, each corresponding to a complete truth assignment to c, r, and u.

[4] $P'(\omega) = P(\omega|\varphi)$ in case φ is observed, and $P'(\omega) = P_\varphi(\omega)$ in case φ is enacted. In both cases $P'(\varphi) = 1$.

[5] In practice, the specification of $U(\omega)$ is done by defining an integer-valued variable V (connoting "value") as a function of a select set of atomic variables. W_i^+ would correspond then to the assertion $V = +i$, $i = 0, 1, 2, \ldots$.

To express our belief that rain does not normally occur in a clear day, we assign a κ value of 1 (indicating one unit of surprise) to any world satisfying $r \wedge \neg c$ and a κ value of 0 to all other worlds (indicating a serious possibility that any such world may be realized). To express the fear of finding ourselves in the rain without an umbrella, we assign a μ value of -1 to worlds satisfying $r \wedge \neg u$ and a μ value of 0 to all other worlds. Thus, $W^+ = false$, $W^0 = \neg(r \wedge \neg u)$, and $W^- = r \wedge \neg u$.

In this simple example, there is no difference between $\kappa_A(\omega)$ and $\kappa(\omega|A)$ because the act A = "Taking an umbrella" has the same implications as the finding "Having an umbrella". Thus, to evaluate the two expressions in Eq. (14), with $A = u$ and $C = c$, we first note that

$$\kappa(W^-|u,c) = \kappa(r \wedge \neg u|u,c) = \infty$$

$$\kappa(W^- \vee W^+|u,c) = \infty$$

so

$$\mu(u; \kappa(\omega|u,c)) = 0$$

Similarly,

$$\kappa(W^-|c) = \kappa(r \wedge \neg u|c) = 0$$

hence

$$\mu(c; \kappa(\omega|c)) = -1 \qquad (15)$$

Thus, $O(u|c)$ is assertable according to the criterion of Eq. (14).

Note that although $\kappa(\omega)$ does not assume that normally we do not have an umbrella with us ($\kappa(u) > 0$), the advice to take an umbrella is still assertable, since leaving u to pure chance might result in harsh consequences (if it rains).

Using the same procedure, it is easy to show that the example also sanctions the assertability of $O(\neg r|c, \neg u)$, which stands for "If it is cloudy and you don't have an umbrella, then you ought to undo (or stop) the rain". This is certainly useless advice, as it does not take into account one's inability to control the weather. Controllability information is not encoded in the ranking functions κ and μ; it should be part of one's causal theory and can be encoded in the language of causal networks using costly preconditions that, until satisfied, would forbid the action $do(A)$ from having any effect on A.[6]

5 COMBINING ACTIONS AND OBSERVATIONS

In this section we develop a probabilistic account for the term $\kappa_A(\omega|C)$, which stands for the belief ranking

[6] In decision theory it is customary to attribute direct costs to actions, which renders $\mu(\omega)$ action-dependent. An alternative, which is more convenient when actions are not enumerated explicitly, is to attribute costs to preconditions that must be satisfied before (any) action becomes effective.

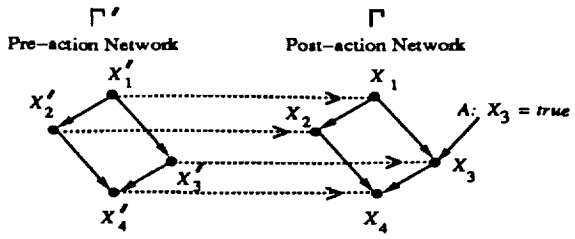


Figure 1: Persistence interactions between two causal networks

that would prevail if we act A after observing C, i.e., the A-update of $k(\omega|C)$. First we note that this update cannot be obtained by simply applying the update formula developed in (Eq. (2.2), Goldszmidt & Pearl 1992),

$$\kappa_A(\omega) = \begin{cases} \kappa(\omega) - \kappa(A|\mathbf{pa}_A(\omega)) & \omega \models A \\ \infty & \omega \models \neg A \end{cases} \qquad (16)$$

where $\mathbf{pa}_A(\omega)$ are the parents (or immediate causes) of A in the causal network Γ evaluated at ω. The formula above was derived under the assumption that Γ is not loaded with any observations (e.g., C) and renders $\kappa_A(\omega)$ undefined for worlds ω that are excluded by previous observations and reinstated by A.

To motivate the proper transformation from $\kappa(\omega)$ to $\kappa_A(\omega|C)$, we consider two causal networks, Γ' and Γ respectively representing the agent's epistemic states before and after the action (see Figure 1). Although the structures of the two networks are almost the same (Γ contains additional root nodes representing the action $do(A)$), it is the interactions between the corresponding variables that determine which beliefs are going to persist in Γ and which are to be "clipped" by the influence of action A.

Let every variable X_i' in Γ' be connected to the corresponding variable X_i in Γ by a directed link $X_i' \longrightarrow X_i$ that represents persistence by default, namely, the natural tendency of properties to persist, unless there is a cause for a change. Thus, the parent set of each X_i in Γ has been augmented with one more variable: X_i'. To specify the conditional probability of X_i, given its new parent set $\{\mathbf{pa}_{X_i} \cup X_i'\}$, we need to balance the tendency of X_i to persist (i.e., be equal to X_i') against its tendency to obey the causal influence exerted by $\mathbf{pa}_{X_i}$. We will assume that persistence forces yield to causal forces and will perpetuate only those properties that are not under any causal influence to terminate. In terms of ranking functions, this assumption reads:

$$\kappa(X_i(\omega)|\mathbf{pa}_i(\omega),\ X_i'(\omega')) =$$

$$\begin{cases} s_i & \text{if } \mathbf{pa}_i = \emptyset \text{ and } X_i(\omega) \neq X_i(\omega') \\ s_i + \kappa(X_i(\omega)|\mathbf{pa}_i(\omega)) & \text{if } X_i(\omega) \neq X_i'(\omega') \text{ and } \kappa(\neg X_i(\omega)|\mathbf{pa}_i(\omega)) = 0 \\ \kappa(X_i(\omega)|\mathbf{pa}_i(\omega)) & \text{otherwise} \end{cases} \qquad (17)$$

where ω' and ω specify the truth values of the variables in the corresponding networks, Γ' and Γ, and $s_i \geq 1$ is

a constant characterizing the tendency of X_i to persist. Eq. (17) states that the past value of X_i may affect the normal relation between X_i and its parents only when it differs from the current value and, at the same time, the parents of X_i do not compel the change. In such a case, the inequality $X_i(\omega) \neq X_i'(\omega')$ contributes s_i units of surprise to the normal relation between X_i and its parents.[7] The unique feature of this model, unlike the one proposed in (Goldszmidt & Pearl 1992), is that persistence defaults can be violated by causal factors without forcing us to conclude that such factors are abnormal.

Eq. (17) specifies the conditional rank $\kappa(X|\mathbf{pa}_X)$ for every variable X in the combined networks and, hence, it provides a complete specification of the joint rank $\kappa(\omega,\omega')$.[8] The desired expression for the post-action ranking $\kappa_A(\omega)$ can then be obtained by marginalizing $\kappa(\omega,\omega')$ over ω':

$$\kappa_A(\omega) = \min_{\omega'} \kappa(\omega,\omega') \tag{18}$$

We need, however, to account for the fact that some variables in network Γ are under the direct influence of the action A, and hence the parents of these nodes are replaced by the action node $do(A)$. If A consists of a conjunction of atomic propositions, $A = \wedge_{j\in J} A_j$, where each A_j stands for either $X_j = true$ or $X_j = false$, then each X_i, $i \in J$, should be exempt from incurring the spontaneity penalty specified in Eq. (17). Additionally, in calculating $\kappa(\omega,\omega')$ we need to sum $\kappa(X_i(\omega)|\mathbf{pa}_i(\omega), X_i'(\omega'))$ only over $i \notin J$, namely, over variables not under the direct influence of A. Thus, collecting terms and writing $\kappa(\omega) = \sum_i \kappa(X_i(\omega)|\mathbf{pa}_i(\omega))$, we obtain

$$\kappa_A(\omega|C) = \kappa(\omega) - \sum_{i\in J\cup R} \kappa(X_i(\omega)|\mathbf{pa}_i(\omega)) + \min_{\omega'}[\sum_{i\notin J} S_i(\omega,\omega') + \kappa(\omega'|C)] \tag{19}$$

where R is the set of root nodes and

$$S_i(\omega,\omega') = \begin{cases} s_i & \text{if } X_i(\omega) \neq X_i(\omega') \text{ and } \mathbf{pa}_i = \emptyset \\ s_i & \text{if } X_i(\omega) \neq X_i(\omega'), \mathbf{pa}_i \neq \emptyset \text{ and} \\ & \quad \kappa(\neg X_i(\omega)|\mathbf{pa}_i(\omega)) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

[7] This is essentially the persistence model used by Dean and Kanazawa (Dean & Kanazawa 1989), in which s_i represents the survival function of X_i. The use of ranking functions allows us to distinguish crisply between changes that are causally supported, $\kappa(\neg X_i(\omega)|\mathbf{pa}_i(\omega)) > 0$, and those that are unsupported, $\kappa(\neg X_i(\omega)|\mathbf{pa}_i(\omega)) = 0$.

[8] The expressions, familiar in probability theory,

$$P(\omega,\omega') = \prod_j P(X_j(\omega,\omega')|\mathbf{pa}_j(\omega,\omega')),\ P(\omega) = \sum_{\omega'} P(\omega,\omega')$$

translate into the ranking expressions

$$\kappa(\omega,\omega') = \sum_j \kappa(X_j(\omega,\omega'))|\mathbf{pa}_j(\omega,\omega')),\ \kappa(\omega) = \min_{\omega'} \kappa(\omega,\omega')$$

where j ranges over all variables in the two networks.

Eq. (19) demonstrates that the effect of observations and actions can be computed as an updating operation on epistemic states, these states being organized by a fixed causal network, with the only varying element being κ, the belief ranking. Long streams of observations and actions could therefore be processed as a sequence of updates on some initial state, without requiring analysis of long chains of temporally indexed networks, as in Dean and Kanazawa (1989).

To handle disjunctive actions such as "Paint the wall either red or blue" one must decide between two interpretations: "Paint the wall red or blue regardless of its current color" or "Paint the wall either red or blue but, if possible, do not change its current color" (see Katsuno & Mendelzon 1991 and Goldszmidt & Pearl 1992). We will adopt the former interpretation, according to which "$do(A \vee B)$" is merely a shorthand for "$do(A) \vee do(B)$". This interpretation is particularly convenient for ranking systems, because for any two propositions, A and B, we have

$$\kappa(A \vee B) = \min[\kappa(A); \kappa(B)] \tag{21}$$

Thus, if we do not know which action, A or B, will be implemented but consider either to be a serious possibility, then

$$\kappa_{A\vee B}(\omega) = \min[\kappa_A(\omega); \kappa_B(\omega)] \tag{22}$$

Accordingly, if A is a disjunction of actions, $A = \bigvee_l A^l$, where each A^l is a conjunction of atomic propositions, then

$$\kappa_A(\omega|C) = \min_l \kappa_{A^l}(\omega|C) \tag{23}$$

Example 2

To demonstrate the interplay between actions and observations, we will test the assertability of the following dialogue:
Robot 1: It is too dark in here.
Robot 2: Then you ought to push the switch up.
Robot 1: The switch is already up.
Robot 2: Then you ought to push the switch down.

The challenge would be to explain the reversal of the "ought" statement in response to the new observation "The switch is already up". The inferences involved in this example revolve around identifying the type of switch Robot 1 is facing, that is whether it is normal (n) or abnormal ($\neg n$) (a normal switch is one that should be pushed up (u) to turn the light on (l)). The causal network, shown in Figure 2, involves three variables:

L - the current state of the light (l vs $\neg l$),
S - the current position of the switch (u vs $\neg u$), and
T - the type of switch at hand (n vs $\neg n$).

The variable L stands in functional relationship to S and T, via

$$l = (n \wedge u) \vee (\neg n \wedge \neg u) \tag{24}$$

or, equivalently, $k = \infty$ unless l satisfies the relation above.

Since initially the switch is believed to be normal, we set $\kappa(\neg n) = 1$, resulting in the following initial ranking:

S	T	L	$\kappa(\omega)$
u	n	l	0
$\neg u$	n	$\neg l$	0
u	$\neg n$	$\neg l$	1
$\neg u$	$\neg n$	l	1

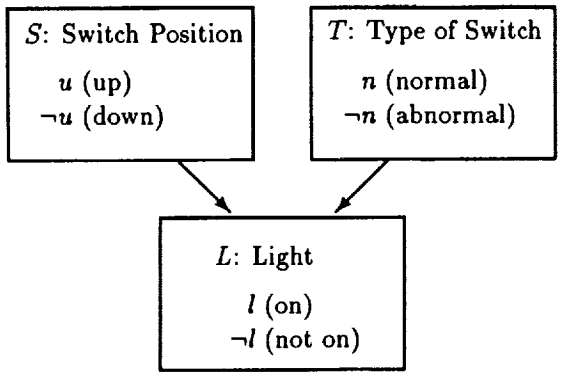


Figure 2: Causal network for Example 2

We also assume that Robot 1 prefers light to darkness, by setting

$$\mu(\omega) = \begin{cases} -1 & \text{if } \omega \models \neg l \\ 0 & \text{if } \omega \models l \end{cases} \quad (25)$$

The first statement of Robot 1 expresses an observation $C = \neg l$, yielding

$$\kappa(\omega|C) = \begin{cases} 0 & \text{for } \omega = \neg u \wedge n \wedge \neg l \\ 1 & \text{for } \omega = u \wedge \neg n \wedge \neg l \\ \infty & \text{for all other worlds} \end{cases} \quad (26)$$

To evaluate $\kappa_A(\omega|C)$ for $A = u$, we now invoke Eq. (19), using the spontaneity functions

$$\begin{aligned} S_T(\omega,\omega') &= 1 \text{ if } T(\omega) \neq T(\omega') \\ S_L(\omega,\omega') &= 0 \text{ if } L(\omega) \neq L(\omega') \end{aligned} \quad (27)$$

because $L(\omega)$, being functionally determined by $\mathbf{pa}_L(\omega)$ is exempt from conforming to persistence defaults. Moreover, for action $A = u$ we also have $\kappa(u|\mathbf{pa}_A) = \kappa(u) = 0$, hence

$$\kappa_A(\omega|C) = \kappa(\omega) - \kappa(T(\omega)) \min_{\omega'=\omega'_1,\omega'_2} \{I[T(\omega) \neq T(\omega')] + \kappa(\omega'|C)\}, \quad \text{for } \omega = \omega_1, \omega_2 \quad (28)$$

where $I[p]$ equals 1 (or 0) if p is *true* (or *false*), and

$$\begin{aligned} \omega_1 &= u \wedge n \wedge l & \omega'_1 &= \neg u \wedge n \wedge \neg l \\ \omega_2 &= u \wedge \neg n \wedge \neg l & \omega'_2 &= u \wedge \neg n \wedge \neg l \end{aligned} \quad (29)$$

All other worlds are excluded by either $A = u$ or $C = \neg l$.

Minimizing Eq. (19) over the two possible ω' worlds, yields

$$\kappa_A(\omega|C) = \begin{cases} 0 & \text{for } \omega = \omega_1 \\ 1 & \text{for } \omega = \omega_2 \end{cases} \quad (30)$$

We see that $\omega_2 = u \wedge \neg n \wedge \neg l$ is penalized with one unit of surprise for exhibiting an unexplained change in switch type (initially believed to be normal).

It is worth noting how ω_1, which originally was ruled out (with $\kappa = \infty$) by the observation $\neg l$, is suddenly reinstated after taking the action $A = u$. In fact, Eq. (19) first restores all worlds to their original $\kappa(\omega)$ value and then adjusts their value in three steps. First it excludes worlds satisfying $\neg A$, then adjusts the $\kappa(\omega)$ of the remaining worlds by an amount $\kappa(A|\mathbf{pa}_A(\omega))$, and finally makes an additional adjustment for violation of persistence.

From Eqs. (26) and (28), we see that $\kappa_A(l|C) = 0 < \kappa(l|C) = \infty$, hence the action $A = u$ meets the assertability criterion of Eq. (14) and the first statement, "You ought to push the switch up", is justified. At this point, Robot 2 receives a new piece of evidence: $S = u$. As a result, $\kappa(\omega|\neg l)$ changes to $\kappa(\omega|\neg l, u)$ and the calculation of $\kappa_A(\omega|C)$ needs to be repeated with a new set of observations, $C = \neg l \wedge u$. Since $\kappa(\omega'|\neg l, u)$ permits only one possible world $\omega' = u \wedge \neg n \wedge \neg l$, the minimization of Eq. (19) can be skipped, yielding (for $A = \neg u$)

$$\kappa_A(\omega|C) = \begin{cases} 0 \text{ for } w = \neg u \wedge \neg n \wedge l \\ 1 \text{ for } w = \neg u \wedge n \wedge \neg l \end{cases} \quad (31)$$

which, in turn, justifies the opposite "ought" statement ("Then you ought to push the switch down"). Note that although finding a normal switch is less surprising than finding an abnormal switch, a spontaneous transition to such a state would violate persistence and is therefore penalized by obtaining a κ of 1.

6 RELATIONS TO OTHER ACCOUNTS

6.1 DEONTIC AND PREFERENCE LOGICS

Ought statements of the pragmatic variety have been investigated in two branches of philosophy, deontic logic and preference logic. Surprisingly, despite an intense effort to establish a satisfactory account of "ought" statements (Von Wright 1963, Van Fraassen 1973, Lewis 1973), the literature of both logics is loaded with paradoxes and voids of principle. This raises the question of whether "ought" statements are destined to forever elude formalization or that the approach taken by deontic logicians has been misdirected. I believe the answer involves a combination of both.

Philosophers hoped to develop deontic logic as a separate branch of conditional logic, not as a synthetic amalgam of logics of belief, action, and causation. In other words, they have attempted to capture the meaning of "ought" using a single modal operator $O(\cdot)$, instead of exploring the couplings between "ought" and other modalities, such as belief, action, causation, and desire. The present paper shows that such an isolationistic strategy has little chance of succeeding. Whereas one can perhaps get by without explicit reference to desire, it is absolutely necessary to have both probabilistic knowledge about the effect of observations on the likelihood of events and causal knowledge about actions and their consequences.

We have seen in Section 3 that to ratify the sentence "Given C, you ought to *do* A", we need to know not merely the relative desirability of the worlds delineated by the propositions $A \wedge C$ and $\neg A \wedge C$, but also the feasibility or likelihood of reaching any one of those worlds in the future, after making our choice of A. We also saw that this likelihood depends critically on how C is confirmed, by observation or by action. Since this information cannot be obtained from the logical content of A and C, it is not surprising that "almost every principle which has been proposed as fundamental to a preference logic has been rejected by some other source" (Mullen 1979).

In fact, the decision theoretic account embodied in Eq. (14) can be used to generate counterexamples to most of the principles suggested in the literature, simply by selecting a combination of κ, μ, and Γ that defies the proposed principle. Since any such principle must be valid in all epistemic states and since we have enormous freedom in choosing these three components, it is not surprising that only weak principles, such as $O(A|C) \Longrightarrow \neg O(\neg A|C)$, survive the test. Among the few that do survive, we find the sure-thing principle:

$$O(A|C) \wedge O(A|\neg C) \Longrightarrow O(A) \qquad (32)$$

read as "If you ought to do$A$ given C and you ought to do$A$ given $\neg C$, then you ought to do A without examining C". But one begins to wonder about the value of assembling a logic from a sparse collection of such impoverished survivors when, in practice, a full specification of κ, μ, and Γ would be required.

6.2 COUNTERFACTUAL CONDITIONALS

Stalnaker (1972) was the first to make the connection between actions and counterfactual statements, and he proposed using the probability of the counterfactual conditional (as opposed to the conditional probability, which is more appropriate for indicative conditionals) in the calculation of expected utilities. Stalnaker's theory does not provide an explicit connection between subjunctive conditionals and causation, however. Although the selection function used in the Stalnaker-Lewis nearest-world semantics can be thought of as a generalization of, and a surrogate for, causal knowledge, it is *too* general, as it is not constrained by the basic features of causal relationships such as asymmetry, transitivity, and complicity with temporal order. To the best of my knowledge, there has been no attempt to translate causal sentences into specifications of the Stalnaker-Lewis selection function.[9] Such specifications were partially provided in (Goldszmidt & Pearl 1992), through the imaging function $\omega^*(\omega)$, and are further refined in this paper by invoking the persistence model (Eq. (19)). Note that a directed acyclic graph is the only ingredient one needs to add to the traditional notion of *epistemic state* so as to specify a causality-based selection function.

From this vantage point, our calculus provides, in essence, a new account of subjunctive conditionals that is more reflective of those used in decision making. The account is based on giving the subjunctive the following causal interpretation: "Given C, if I were to perform A, then I believe B would come about", written $A > B|C$, which in the language of ranking function reads

$$\kappa(\neg B|C) = 0 \text{ and } \kappa_A(\neg B|C) > 0 \qquad (33)$$

The equality states that $\neg B$ is considered a serious possibility prior to performing A, while the inequality renders $\neg B$ surprising after performing A. This account, which we call *Decision Making Conditionals* (DMC), avoids several paradoxes of conditional logics (see Nute 1992) and is further described in (Pearl 1993).

6.3 OTHER DECISION THEORETIC ACCOUNTS

Poole (1992) has proposed a quantitative decision-theoretic account of defaults, taking the utility of A, given evidence e, to be

$$\mu(A|e) = \Sigma_\omega \, \mu(\omega, A) P(\omega|e) \qquad (34)$$

This requires a specification of an action-dependent preference function for each (ω, A) pair. Our proposal (in line with (Stalnaker 1972)) attributes the dependence of μ on A to beliefs about the possible consequences of A, thereby keeping the utility of each consequence constant. In this way, we see more clearly how the structure of causal theories should affect the choice of actions. For example, suppose A and e are incompatible ("If the light is on (e), turn it off (A)"), taking (34) literally (without introducing temporal indices) would yield absurd results. Additionally, Poole's is a calculus of incremental improvements of utility, while

[9] Gibbard and Harper (Gibbard & Harper 1980) develop a quantitative theory of rational decisions that is based on Stalnaker's suggestion and explicitly attributes causal character to counterfactual conditionals. However, they assume that probabilities of counterfactuals are given in advance and do not specify either how such probabilities are encoded or how they relate to probabilities of ordinary propositions. Likewise, a criterion for accepting a counterfactual conditional, given other counterfactuals and other propositions, is not provided.

ours is concerned with substantial improvements, as is typical of ought statements.

Boutilier (1993) has developed a modal logic account of conditional goals which embodies considerations similar to ours. It remains to be seen whether causal relationships such as those governing the interplay among actions and observations can easily be encoded into his formalism.

7 CONCLUSION

By pursuing the semantics of ought statements this paper develops an account of qualitative decision theory and a framework for qualitative planning under uncertainty. The two main features of this account are:
1. Order-of-magnitude specifications of probabilities and utilities are combined to produce qualitative expected utilities of actions and consequences, conditioned on observations (Eq. (7)).
2. A single causal network, combined with universal assumptions of persistence is sufficient for specifying the dynamics of beliefs under any sequence of actions and observations (Eq. (9)).

Acknowledgements

This work benefitted from discussions with Craig Boutilier, Adnan Darwiche, Moisés Goldszmidt, and Sek-Wah Tan. The research was partially supported by Air Force grant #AFOSR 90 0136, NSF grant #IRI-9200918, Northrop Micro grant #92-123, and Rockwell Micro grant #92-122.

References

Alchourron, C., Gardenfors, P., and Makinson, D., "On the logic of theory change: Partial meet contraction and revision functions," *Journal of Symbolic Logic*, 50, 510-530, 1985.

Boutilier, C., "A modal characterization of defeasible deontic conditionals and conditional goals," *Working Notes of the AAAI Spring Symposium Series*, Stanford, CA, 30-39, March 1993.

Dean, T., and Kanazawa, K., "A model for reasoning about persistence and causation," *Computational Intelligence*, 5, 142-150, 1989.

Gibbard, A., and Harper, L., "Counterfactuals and two kinds of expected utility," in Harper, L., Stalnaker, R., and Pearce, G. (Eds.), *Ifs*, D. Reidel Dordrecht, Holland, 153-190, 1980.

Goldszmidt, M., "Qualitative probabilities: A normative framework for commonsense reasoning," Technical Report (R-190), UCLA, Ph.D. Dissertation, October 1992.

Goldszmidt, M., and Pearl, J., "Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions," in *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, CA, 661-672, October 1992.

Harper, L., Stalnaker, R., and Pearce, G. (Eds.), *Ifs*, D. Reidel Dordrecht, Holland, 1980.

Katsuno, H., and Mendelzon, A.O., "On the difference between updating a knowledge base and revising it," in *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference*, Morgan Kaufmann, San Mateo, CA, 387-394, 1991.

Lewis, D., *Counterfactuals*, Harvard University Press, Cambridge, MA, 1973.

Nute, D., "Logic, conditional," in Stuart C. Shapiro (Ed.), *Encyclopedia of Artificial Intelligence*, 2nd Edition, John Wiley, New York, 854-860, 1992.

Mullen, J.D., "Does the logic of preference rest on a mistake?", *Metaphilosophy*, 10, 247-255, 1979.

Pearl, J., *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, CA, 1988.

Pearl, J., "From Adams' conditionals to default expressions, causal conditionals, and counterfactuals," UCLA, Technical Report (R-193), 1993. To appear in *Festschrift for Ernest Adams*, Cambridge University Press, 1993.

Poole, D., "Decision-theoretic defaults," *9th Canadian Conference on AI*, Vancouver, Canada, 190-197, May 1992.

Shachter, R.D., "Evaluating influence diagrams," *Operations Research*, 34, 871-882, 1986.

Shoham, Y., *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence*, MIT Press, Cambridge, MA, 1988.

Spohn, W., "Ordinal conditional functions: A dynamic theory of epistemic states," in W.L. Harper and B. Skyrms (Eds.), *Causation in Decision, Belief Change, and Statistics*, D. Reidel, Dordrecht, Holland, 105-134, 1988.

Stalnaker, R., "Letter to David Lewis" in Harper, L., Stalnaker, R., and Pearce, G. (Eds.), *Ifs*, D. Reidel Dordrecht, Holland, 151-152, 1980.

Van Fraassen, B.C., "The logic of conditional obligation," in M. Bunge (Ed.), *Exact Philosophy*, D. Reidel, Dordrecht, Holland, 151-172, 1973.

Von Wright, G.H., *The Logic of Preference*, University of Edinburgh Press, Edinburgh, Scotland, 1963.

N94-34044

Finding Accurate Frontiers: A Knowledge-Intensive Approach to Relational Learning

Michael Pazzani and Clifford Brunk

Department of Information and Computer Science
University of California
Irvine, CA 92717
pazzani@ics.uci.edu, brunk@ics.uci.edu

Abstract

An approach to analytic learning is described that searches for accurate entailments of a Horn Clause domain theory. A hill-climbing search, guided by an information based evaluation function, is performed by applying a set of operators that derive frontiers from domain theories. The analytic learning system is one component of a multi-strategy relational learning system. We compare the accuracy of concepts learned with this analytic strategy to concepts learned with an analytic strategy that operationalizes the domain theory.

Introduction

There are two general approaches to learning classification rules. Empirical learning programs operate by finding regularities among a group of training examples. Analytic learning systems use a domain theory[1] to explain the classification of examples, and form a general description of the class of examples with the same explanation. In this paper, we discuss an approach to learning classification rules that integrates empirical and analytic learning methods. The goal of this integration is to create concept descriptions that are more accurate classifiers than both the original domain theory (which serves as input to the analytic learning component) and the rules that would arise if only the empirical learning component were used. We describe a new analytic learning method that returns a frontier (i.e., conjunctions and disjunctions of operational[2] and non-operational literals) instead of an operationalization (i.e., a conjunction of operational literals) and we demonstrate there is an accuracy advantage in allowing an analytic learner to dynamically select the level of generality of the learned concept, as a function of the training data.

In previous work (Pazzani, et al., 1991; Pazzani & Kibler, 1992), we have described FOCL, a system that extends Quinlan's (1990) FOIL program in a number of ways, most significantly by adding a compatible explanation-based learning (EBL) component. In this paper we provide a brief review of FOIL and FOCL, then discuss how operationalizing a domain theory can adversely affect the accuracy of a learned concept. We argue that instead of operationalizing a domain theory, an analytic learner should return the most general implication of the domain theory, provided this implication is not less accurate than any more specialized implication. We discuss the computational complexity of an algorithm that enumerates all such descriptions and then describe a greedy algorithm that efficiently addresses the problem. Finally, we present a variety of experiments that indicate replacing the operationalization algorithm of FOCL with the new analytic learning method results in more accurate learned concept descriptions.

FOIL

FOIL learns classification rules by constructing a set of Horn Clauses in terms of known operational predicates. Each clause body consists of a conjunction of literals that cover some positive and no negative examples. FOIL starts to learn a clause body by finding the literal with the maximum information gain, and continues to add literals to the clause body until the clause does not cover any negative examples. After learning each clause, FOIL removes from further consideration the positive examples covered by that clause. The learning process ends when all positive examples have been covered by some clause.

FOCL

FOCL extends FOIL by incorporating a compatible EBL component. This allows FOCL to take advantage of an initial domain theory. When constructing a clause body, there are two ways that FOCL can add literals. First, it can create literals via the same empirical method used by FOIL. Second, it can create literals by operationalizing a target concept, i.e., a non-operational definition of the concept to be learned (Mitchell, et al., 1986). FOCL uses FOIL's information-based evaluation function to determine whether to add a literal learned empirically or a conjunction of literals learned analytically. In general FOCL learns clauses of the form $r \leftarrow O_i \wedge O_d \wedge O_f$ where O_i is an initial conjunction of operational literals learned empirically, O_d is a conjunction of literals found by operationalizing the domain theory, and O_f is a final conjunction of literals learned empirically[3]. Pazzani, et al. (1991) demonstrate

1. We use *domain theory* to refer to a set of Horn-Clause rules given to a learner as an approximate definition of a concept and *learned concept* to refer to the result of learning.

2. We use the term *operational* to refer to predicates that are defined *extensionally* (i.e., defined by a collection of facts). However, the results apply to any satirically determined definition of operationality.

3. Note the target concept is operationalized at most once per clause and that either O_i, O_d, or O_f may be empty.

that FOCL can utilize incomplete and incorrect domain theories. We attribute this capability to its uniform use of an evaluation function to decide whether to include literals learned empirically or analytically.

Operationalization in FOCL differs from that of most EBL programs in that it uses a set of positive and negative examples, rather than a single positive example. A non-operational literal is operationalized by producing a specialization of a domain theory that is a conjunction of operational literals. When there are several ways of operationalizing a literal (i.e., there are multiple, disjunctive clauses), the information gain metric is used to determine which clause should be used by computing the number of examples covered by each clause. Figure 1 displays a typical domain theory with an operationalization (fAgAhAkAlApAq) represented as bold nodes.

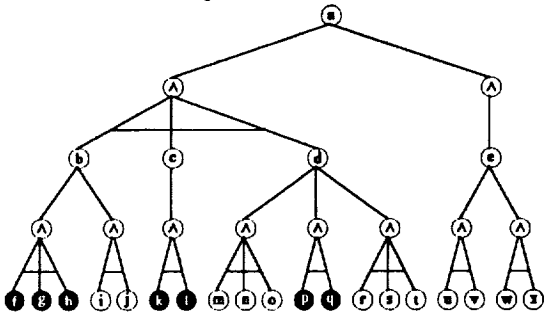


Figure 1. The bold nodes represent one operationalization (fAgAhAkAlApAq) of the domain theory. In standard EBL, this path would be chosen if it were a proof of a single positive example. In FOCL, this path would be taken if the choice made at a disjunctive node had greater information gain (with respect to a set of positive and negative examples) than alternative choices.

Operationalization

The operationalization process yields a specialization of the target concept. Indeed, several systems designed to deal with overly general theories rely on the operationalization process to specialize domain theories (Flann & Dietterich, 1990; Cohen, 1992). However, fully operationalizing a domain theory can result in several problems:

1. Overspecialization of correct non-operational concepts. For example, if the domain theory in Figure 1 is completely correct, then a correct operational definition will consist of eight clauses. However, if there are few examples, or some combinations of operationalizations are rare, then there may not be a positive example corresponding to all combinations of all operationalizations of non-operational predicates. As a consequence, the learned concept may not include some combinations of operational predicates (e.g., iAjAkAlArAsAt), although there is no evidence that these specializations are incorrect.
2. Replication of empirical learning. If there is a literal omitted from a clause of a non-operational predicate, then this literal will be omitted from each operationalization involving this predicate. For example, if the domain theory in Figure 1 erroneously contained the rule b←fAh instead of b←fAgAh, then each operationalization of the target concept using this predicate (i.e., fAhAkAlAmAnAo, fAhAkAlApAq, and fAhAkAlArAsAt) will contain the same omission. FOCL can recover from this error if its empirical component can find the omitted literal, g. However, to obtain a correct learned concept description, FOCL would have to find the same condition independently three times on three different sets of examples. This replication of empirical learning is analogous to the replicated subtree problem in decision trees (Pagallo & Haussler, 1990). This problem should be most noticeable when there are few training examples. Under this circumstance, it is unlikely that empirical learning on several arbitrary partitions of a data set will be as accurate as learning from the larger data set.
3. Proofs involving incorrect non-operational predicates may be ignored. If the definition of a non-operational predicate (e.g., c in Figure 1) is not true of any positive example, then the analytic learner will not return any operationalization using this predicate. This reduces the usefulness of the domain theory for an analytic learner. For example, if c is not true of any positive example, then FOCL as previously described can find only two operationalizations: uAv and wAx. Again, we anticipate that this problem will be most severe when there are few training examples. With many examples, the empirical learner can produce accurate clauses that mitigate this problem.

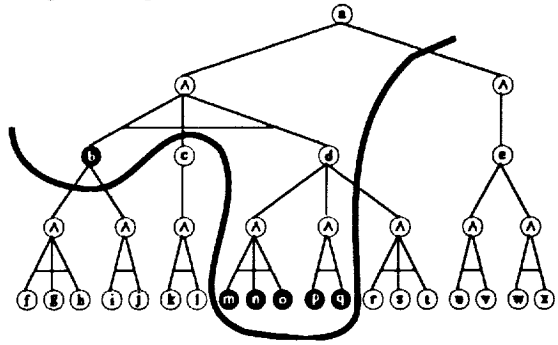


Figure 2. The bold nodes represent one frontier of the domain theory, bA((mAnAo)v(pAq)).

Frontiers of a Domain Theory

To address the problems raised in the previous section, we propose an analytic learner that does not necessarily fully operationalize target concepts. Instead, the learner returns a *frontier* of the domain theory. A frontier differs from an operationalization of a domain theory in three ways. The frontier represented by those nodes immediately above the line in Figure 2, bA((mAnAo)v(pAq)), illustrates these differences:

1. Non-operational predicates (e.g., b) can appear in the frontier.

2. A disjunction of two or more clauses that define a non-operational predicate (e.g., $(m \wedge n \wedge o) \vee (p \wedge q)$) can appear in the frontier.
3. A frontier does not necessarily include all literals in a conjunction (e.g., neither c, nor any specialization of c, appears in the frontier).

Combined, the first two distinguishing features of a frontier address the first two problems associated with operationalization. Overspecialization of correct non-operational concepts can be avoided if the analytic component returns a more general concept description. Similarly, replication of empirical learning can be avoided if the analytic component returns a frontier more general than an operationalization. For example, if the domain theory in Figure 2 erroneously contained the rule $b \leftarrow f \wedge h$ instead of $b \leftarrow f \wedge g \wedge h$ and frontier $f \wedge h \wedge k \wedge l \wedge d$ was returned, then an empirical learner would only need to be invoked once to specialize this conjunction by adding g. Of course, if one of the clauses defining d were incorrect, it would make sense to specialize d. However, operationalization is not the only means of specialization. For example, if the analytic learner returned $f \wedge h \wedge k \wedge l \wedge ((m \wedge n \wedge o) \vee (p \wedge q))$, then replication of induction problem could also be avoided. This would be desirable if the clause $d \leftarrow r \wedge s \wedge t$ were incorrect.

The third problem with operationalization can be addressed by removing some literals from a conjunction. For example, if no positive examples use $a \leftarrow b \wedge c \wedge d$ because c is not true of any positive example, then the analytic learner might want to consider ignoring c and trying $a \leftarrow b \wedge d$. This would allow potentially useful parts of the domain theory (e.g. b and d) to be used by the analytic learner, even though they may be conjoined with incorrect parts.

The notion of a frontier has been used before in analytic learning. However, the previous work has assumed that the domain theory is correct and has focused on increasing the utility of learned concepts (Hirsh, 1988; Keller, 1988; Segre, 1987) or learning from intractable domain theories (Braverman & Russell, 1988). Here, we do not assume that the domain theory is correct.

We argue that to increase the accuracy of learned concepts, an analytic learner should have the ability to select the generality of a frontier derived from a domain theory. To validate our hypothesis, we will replace the operationalization procedure in FOCL with an analytic learner that returns a frontier. In order to avoid confusion with FOCL, we use the name FOCL-FRONTIER to refer to the system that combines this new analytic learner with an empirical learning component based on FOIL. In general, FOCL-FRONTIER learns clauses of the form $r \leftarrow O_i \wedge F_d \wedge O_f$ where O_i is an initial conjunction of operational literals learned empirically, F_d is a frontier of the domain theory, and O_f is a final conjunction of literals learned empirically. We anticipate that due to its use of a frontier rather than an operationalization, FOCL-FRONTIER will be more accurate than FOCL, particularly when there are few training examples or the domain theory is very accurate.

Enumerating Frontiers of a Domain Theory

Formally, a frontier can be defined as follows. Let b represent a conjunction of literals and p represent a single literal.

1. The target concept is a frontier.
2. A new frontier can be formed from an existing frontier by replacing a literal p with $b_1 \vee \ldots \vee b_i \vee \ldots \vee b_n$ provided there are rules $p \leftarrow b_1, \ldots, p \leftarrow b_i, \ldots, p \leftarrow b_n$.
3. A new frontier can be formed from an existing frontier by replacing a disjunction $b_1 \vee \ldots \vee b_{i-1} \vee b_i \vee b_{i+1} \vee \ldots \vee b_n$ with $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_n$ for any i. This deletes b_i.
4. A new frontier can be formed from an existing frontier by replacing a conjunction $p_1 \wedge \ldots \wedge p_{i-1} \wedge p_i \wedge p_{i+1} \wedge \ldots \wedge p_n$ with $p_1 \wedge \ldots \wedge p_{i-1} \wedge p_{i+1} \wedge \ldots \wedge p_n$ for any i. This deletes p_i.

One approach to analytic learning would be to enumerate all possible frontiers. The information gain of each frontier could be computed, and if the frontier with the maximum information gain has greater information gain than any literal found empirically, then this frontier would be added to the clause under construction. Such an approach would be impractical for all but the most trivial, non-recursive domain theories. Since each frontier specifies a unique combination of leaf nodes of an and-or tree (i.e., selecting all leaves of a subtree is equivalent to selecting the root of the subtree and selecting no leaves of a subtree is equivalent to deleting the root of a subtree), there are 2^k frontiers of a domain theory that has k nodes in the and/or tree. For example, if every non-operational predicate has n clauses, each clause is a conjunction of m literals, and inference chains have a depth of d *and*-nodes, then the number of frontiers is $2^{m^d n^d}$.

Deriving Frontiers from the Target Concept

Due to the intractability of enumerating all possible frontiers, we propose a heuristic approach based upon hill-climbing search. The frontier is initialized to the target concept. A set of transformation operators is applied to the current frontier to create a set of possible frontiers. If none of the possible frontiers has information gain greater than that of the current frontier[4], then the current frontier is returned. Otherwise, the potential frontier with the maximum information gain becomes the current frontier and the process of applying transformation operators is repeated. The following transformation operators are used[5]:

- *Clause specialization*:

  If there is a frontier containing a literal p, and there are exactly n rules of the form $p \leftarrow b_1, \ldots, p \leftarrow b_i, \ldots, p \leftarrow b_n$, then n frontiers formed by replacing p with b_i are evaluated.

4. The information gain of a frontier is calculated in the same manner than Quinlan (1990) calculates the information gain of a literal: by counting the number of positive and negative examples that meet the conditions represented by the frontier.

5. The numeric restrictions placed upon the applicability of each operator are for efficiency reasons (i.e., to ensure that each unique frontier is evaluated only once).

- *Specialization by removing disjunctions*:
  a. If there is a frontier containing a literal p, and there are n rules of the form $p \leftarrow b_1, \ldots, p \leftarrow b_i, \ldots, p \leftarrow b_n$, then n frontiers formed by replacing p with $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_n$ are evaluated (provided n>2).
  b. If there is a frontier containing a disjunction $b_1 \vee \ldots \vee b_{i-1} \vee b_i \vee b_{i+1} \vee \ldots \vee b_m$, then m frontiers replacing this disjunction with $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_m$ are evaluated (provided m>2).
- *Generalization by adding disjunctions*:
  If there is a frontier containing a (possibly trivial) disjunction of conjunction of literals $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_m$ and there are rules of the form $p \leftarrow b_1, \ldots, p \leftarrow b_{i-1}, p \leftarrow b_i, p \leftarrow b_{i+1}, \ldots, p \leftarrow b_n$ and m<n-1, then n–m frontiers replacing the disjunction $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_m$ with $b_1 \vee \ldots \vee b_{i-1} \vee b_i \vee b_{i+1} \vee \ldots \vee b_m$ are evaluated. This is implemented efficiently by keeping a derivation of each frontier, rather than by searching for frontiers matching this pattern.
- *Generalization by literal deletion*:
  If there is a frontier containing a conjunction of literals $p_1 \wedge \ldots \wedge p_{i-1} \wedge p_i \wedge p_{i+1} \wedge \ldots \wedge p_n$, then n frontiers replacing this conjunction with $p_1 \wedge \ldots \wedge p_{i-1} \wedge p_{i+1} \wedge \ldots \wedge p_n$ are evaluated.

There is a close correspondence between the recursive definition of a frontier and these transformation operators. However, there is not a one-to-one correspondence because we have found empirically that in some situations it is advantageous to build a disjunction by adding disjuncts and in other cases it is advantageous to build a disjunction by removing disjuncts. The former tends to occur when few clauses of a predicate are correct while the latter tends to occur when few clauses are incorrect.

Note that the first three frontier operators derive logical entailments from the domain theory while the last does not. Deleting literals from a conjunction is a means of finding an abductive hypothesis. For example, in EITHER (Ourston & Mooney, 1990), a literal can be assumed to be true during the proof process of a single example. One difference between FOCL-FRONTIER and the abduction process of EITHER is that EITHER considers all likely assumptions for each unexplained positive example, and FOCL-FRONTIER uses a greedy approach to deletion based on an evaluation of the effect on a set of examples.

Evaluation

In this section, we report on a series of experiments in which we compare FOCL using empirical learning alone (EMPIRICAL), FOCL using a combination of empirical learning and operationalization, and FOCL-FRONTIER. We evaluate the performance of each algorithm in several domains. The goal of these experiments is to substantiate the claim that analytic learning via frontier transformations results in more accurate learned concept descriptions than analytic learning via operationalization. Throughout this paper, we use an analysis of variance to determine if the difference in accuracy between algorithms is significant.

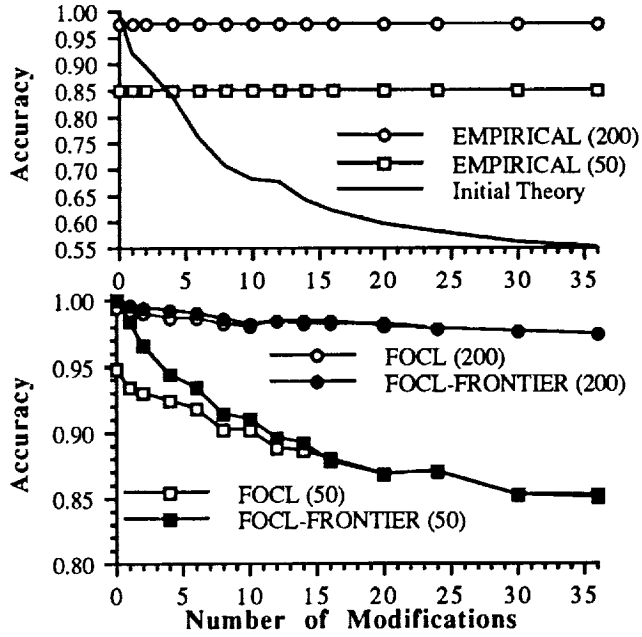


Figure 3. A comparison of FOCL's empirical component (EMPIRICAL), FOCL using both empirical learning and operationalization, and FOCL-FRONTIER in the chess end gain domain. **upper**: The accuracy of EMPIRICAL (given training sets of size 50 and 200) and the average accuracy of the initial theory as a function of the number of changes to the domain theory. **lower**: The accuracy of FOCL and FOCL-FRONTIER on the same data.

Chess End Games

The first problem we investigate is learning rules that determine if a chess board containing a white king, white rook, and black king is in an illegal configuration. This problem has been studied using empirical learning systems by Muggleton, et al. (1989) and Quinlan (1990). Here, we compare the accuracy of FOCL-FRONTIER and FOCL using a methodology identical to that used by Pazzani and Kibler (1992) to compare FOCL and FOIL.

In these experiments the initial theory given to FOCL and FOCL-FRONTIER was created by introducing either 0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 30 or 36 random modifications to a correct domain theory that encodes the relevant rules of chess. Four types of modifications were made: deleting a literal from a clause, deleting a clause, adding a literal to a clause, and adding a clause. Added clauses are constructed with random literals. Each clause contains at least one literal, there is a 0.5 probability that a clause will have at least two literals, a 0.25 probability of containing at least three, and so on.

We ran experiments using 25, 50, 75, 150, and 200 training examples. On each trial the training and test examples were drawn randomly from the set of 8^6 possible board configurations. We ran 32 trials of each algorithm and measured the accuracy of the learned concept description on 1000 examples. For each algorithm the

curves for 50 and 200 training examples are presented. Figure 3 (upper) graphs the accuracy of the initial theory and the concept description learned by FOCL's empirical component as functions of the number of modifications to the correct domain theory. Figure 3 (lower) graphs the accuracy of FOCL and FOCL-FRONTIER.

The following conclusions may be drawn from these experiment. First, FOCL-FRONTIER is more accurate than FOCL when there are few training examples. An analysis of variance indicates that the analytic learning algorithm has a significant effect on the accuracy ($p<.0001$) when there are 25, 50 and 75 training examples. However, where there are 150 or 200 training examples, there is no significant difference in accuracy between the analytic learning algorithms because both analytic learning algorithms (as well as the empirical algorithm) are very accurate on this problem with larger numbers of training examples. Second, the difference in accuracy between FOCL and FOCL-FRONTIER is greatest when the domain theory has few errors. With 25 and 50 examples, there is a significant interaction between the number of modifications to the domain theory and the algorithm ($p<.0001$ and $p<.005$, respectively).

During these experiments, we also recorded the amount of work EMPIRICAL, FOCL and FOCL-FRONTIER performed while learning a concept description. Pazzani and Kibler (1990) argue that the number of times information gain is computed is a good metric for describing the size of the search space explored by FOCL. Figure 4 graphs these data as a function of the number of modifications to the domain theory for learning with 50 training examples. FOCL-FRONTIER tests only a small percentage of the 225 frontiers of this domain theory with 25 leaf nodes. The frontier approach requires less work than operationalization until the domain theory is fairly inaccurate. This occurs, in spite of the larger branching factor because the frontier approach generates more general concepts with fewer clauses than those created by operationalization (see Table 1). When the domain theory is very inaccurate, FOCL and FOCL-FRONTIER perform slightly more work than EMPIRICAL because there is a small overhead in determining that the domain theory has no information gain.

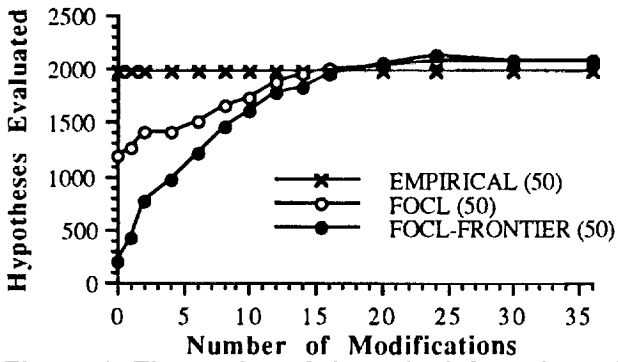


Figure 4: The number of times the information gain metric is computed for each algorithm.

FOCL (92.6% accurate)

```
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKf,WRf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKr,WRr).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←near(WKr,BKr) ∧
                                  near(WKf,BKf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKr,WKf) ∧
                                  equal(WKr,BKr) ∧
                                  near(WKf,BKf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(WKr,WRr) ∧
                                  equal(WKf,WRf).
```

FOCL-FRONTIER (98.3% accurate)

```
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←k_attack(WKr,WKf,BKr,BKf) ∨
                                  r_attack(WRr,WRf,BKr,BKf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKf,WRf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←same_pos(WKr,WKf,WRr,WRf).
```

Table 1. Typical definitions of `illegal`. The variables refer to the rank and file of the white king, white rook, and the black king. The domain theory was 91.0% accurate and 50 training examples were used.

Educational Loans

The second problem studied involves determining if a student is required to pay back a loan based on enrollment and employment information. This theory was constructed by an honors student who had experience processing loans. This problem, available from the UC Irvine repository, was previously used by an extension to FOCL that revises domain theories (Pazzani & Brunk, 1991). The domain theory is 76.8% accurate on a set of 1000 examples.

We ran 16 trials of FOCL and FOCL-FRONTIER with this domain theory on randomly selected training sets ranging from 10 to 100 examples and measured the accuracy of the learned concept by testing on 200 distinct test examples. The results indicate that the learning algorithm has a significant effect on the accuracy of the learned concept ($p<.0001$). Figure 5 plots the mean accuracy of the three algorithms as a function of the number of training examples.

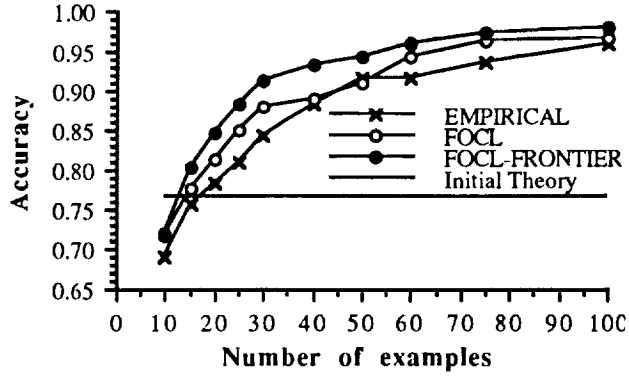


Figure 5. The accuracy of FOCL's empirical component alone, FOCL with operationalization and FOCL-FRONTIER on the student loan data.

Nynex Max

Nynex Max (Rabinowitz, et al., 1991) is an expert system that is used by NYNEX (the parent company of New York Telephone and New England Telephone) at several sites to determine the location of a malfunction for customer-reported telephone troubles. It can be viewed as solving a

classification problem where the input is data such as the type of switching equipment, various voltages and resistances and the output is the location to which a repairman should be dispatched (e.g., the problem is in the customer's equipment, the customer's wiring, the cable facilities, or the central office). Nynex Max requires some customization at each site in which it is installed.

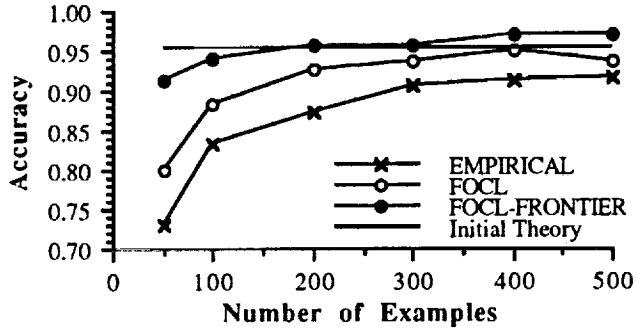


Figure 6. The accuracy of the learning algorithms at customizing the Max knowledge-base.

In this experiment, we compare the effectiveness of FOCL-FRONTIER and FOCL at customizing the Nynex Max knowledge-base. The initial domain theory is taken from one site, and the training data is the desired output of Nynex Max at a different site. Figure 6 shows the accuracy of the learning algorithms (as measured on 200 independent test examples), averaged over 10 runs as a function of the number of training examples. FOCL-FRONTIER is more accurate than FOCL ($p<.0001$). This occurs because the initial domain theory is fairly large (about 75 rules), very disjunctive, and fairly accurate (about 95.4%). Under these circumstances, FOCL requires many examples to form many operational rules, while FOCL-FRONTIER learns fewer, more general rules. FOCL-FRONTIER is the only algorithm to achieve an accuracy significantly higher than the initial domain theory.

Related Work

Cohen (1990; 1991a) describes the ELGIN systems that makes use of background knowledge in a way similar to FOCL-FRONTIER. In particular, one variant of ELGIN called ANA-EBL, finds concepts in which all but k nodes of a proof tree are operational. The algorithm, which is exponential in k, learns more accurate rules from overly general domain theories than an algorithm that uses only operational predicates. A different variant of ELGIN, called K-TIPS, selects k nodes of a proof tree and returns the most general nodes in the proof tree that are not ancestors of the selected nodes. This enables the system to learn a set of clauses containing at most k literals from the proof tree. Some of the literals may be non-operational and some subtrees may be deleted from the proof tree. In some ways, ELGIN is like the optimal algorithm we described above that enumerates all possible frontiers. A major difference is that ELGIN does not allow disjunction in proofs, and for efficiency reasons is restricted to using small values of k. FOCL-FRONTIER is not restricted in such a fashion, since it relies on hill-climbing search to avoid enumerating all possible hypotheses. In addition, the empirical learning component of FOCL-FRONTIER allows it to learn from overly specific domain theories in addition to overly general domain theories.

In the GRENDEL system, Cohen (1991b) uses a grammar rather than a domain theory to generate hypotheses. Cohen shows that this grammar provides an elegant way to describe the hypothesis space searched by FOCL. It is possible to encode the domain theory in such a grammar. In addition, it is possible to encode the hypothesis space searched by FOIL in the grammar. GRENDEL uses a hill-climbing search method similar to the operationalization process in FOCL to determine which hypothesis to derive from the grammar. Cohen (1991b) shows that augmenting GRENDEL with advice to prefer grammar rules corresponding to the domain theory results in concepts that are as accurate as those of FOCL (with operationalization) on the chess end game problem. The primary difference between GRENDEL and FOCL-FRONTIER is that FOCL-FRONTIER contains operators for deleting literals from and-nodes and for incorporating several disjunctions from or-nodes. However, due to the generality of GRENDEL's grammatical approach, it should be possible to extend GRENDEL by writing a preprocessor that converts a domain theory into a grammar that simulate these operators. Here, we have shown that these operators result in increased accuracy, so it is likely that a grammar based on the operators proposed here would increase GRENDEL's accuracy.

FOCL-FRONTIER is in some ways similar to theory revision systems, like EITHER (Ourston & Mooney, 1990). However, theory revision systems have an additional goal of making minimal revisions to a theory, while FOCL-FRONTIER uses a set of frontiers from the domain theory (and/or empirical learning) to discriminate positive from negative examples. EITHER deals with propositional theories and would not be able to revise any of the relational theories used in the experiments here. A more recent theory revision system, FORTE (Richards & Mooney, 1991), is capable of revising relational theories. It has been tested on one problem on which we have run FOCL, the illegal chess problem from Pazzani & Kibler (1992). Richards (1992) reports that with 100 training examples FOCL is significantly more accurate than FORTE (97.9% and 95.6% respectively). For this problem, FOCL-FRONTIER is 98.5% accurate (averaged over 20 trials). FORTE has a problem with this domain, since it contains two overly-general clauses for the same relation and its revision operators assume that at most one clause is overly general. Although it is not possible to draw a general conclusion form this single example, it does indicate that there are techniques for taking advantage of information contained in a theory that FOCL utilizes that are not incorporated into FORTE.

Future Work

Here, we have described one set of general purpose operators that derive frontiers. We are currently experimenting with more special purpose operators designed to handle commonly occurring problems in knowledge-based systems. For example, one might wish to consider operators that negate a literal in a frontier (since we occasionally omit a not from rules) or that change the order of arguments to a predicate. Initial experiments (Pazzani, 1992) with one such operator in FOCL (replacing one predicate with a related predicate) yielded promising results.

Conclusion

In this paper, we have presented an approach to integrating empirical and analytic learning that differs from previous approaches in that it uses an information theoretic metric on a set of training examples to determine the generality of the concepts derived from the domain theory. Although it is possible that the hill-climbing search algorithm will find a local maximum, experimentally we have demonstrated that in situations where there are few training examples, the domain theory is very accurate, or the domain theory is highly disjunctive this approach learns more accurate concept descriptions than either empirical learning alone or a similar approach that integrates empirical learning and operationalization. From this we conclude that there is an advantage in allowing the analytic learner to select the generality of a frontier derived from a domain theory both in terms of accuracy and in terms of the amount of work required to learn a concept description.

Acknowledgments

This research is supported by an Air Force Office of Scientific Research Grant, F49620-92-J-0430, and by the University of California, Irvine through an allocation of computer time. We thank Kamal Ali, William Cohen, Andrea Danyluk, Caroline Ehrlich, Dennis Kibler, Ray Mooney and Jim Wogulis for comments on an earlier draft of this paper.

References

Braverman, M. & Russell, S. (1988). Boundaries of operationality. Proceedings of the Fifth International Conference on Machine Learning (pp. 221–233). Ann Arbor, MI: Morgan Kaufmann.

Cohen, W. (1990). Explanation-based generalization as an abstraction mechanism in concept learning. Technical Report DCS-TR-271 (Ph.D. dissertation). Rutgers University.

Cohen, W. (1991a). The generality of overgenerality. Proceedings of the Eighth International Workshop on Machine Learning (pp. 490–494). Evanston, IL: Morgan Kaufmann.

Cohen, W. (1991b). Grammatically biased learning: Learning Horn Theories using an explicit antecedent description language. AT&T Bell Laboratories Technical Report 11262-910708-16TM (available from the author).

Cohen, W. (1992). Abductive explanation-based learning: A solution to the multiple inconsistent explanation problem. Machine Learning.

Flann, N., & Dietterich, T. (1990). A study of explanation-based methods for inductive learning. Machine Learning, 4, 187–226.

Hirsh, H. (1988). Reasoning about operationality for explanation-based learning. Proceedings of the Fifth International Conference on Machine Learning (pp. 214–220). Ann Arbor, MI: Morgan Kaufmann.

Keller, R. (1988). Operationality and generality in explanation-based learning: Separate dimensions or opposite end-points. AAAI Spring Symposium on Explanation-Based Learning. Stanford University.

Mitchell, T., Keller, R., & Kedar-Cabelli, S. (1986). Explanation-based learning: A unifying view. Machine Learning, 1, 47–80.

Muggleton, S., Bain, M., Hayes-Michie, J., & Michie, D. (1989). An experimental comparison of human and machine learning formalisms. Proceedings of the Sixth International Workshop on Machine Learning (pp. 113–118). Ithaca, NY: Morgan Kaufmann.

Ourston, D., & Mooney, R. (1990). Changing the rules: A comprehensive approach to theory refinement. Proceedings of the Eighth National Conference on Artificial Intelligence (pp. 815–820). Boston, MA: Morgan Kaufmann.

Pagallo, G., & Haussler, D. (1990). Boolean feature discovery in empirical learning. Machine Learning, 5, 71–100.

Pazzani, M., & Brunk, C. (1991). Detecting and correcting errors in rule-based expert systems: an integration of empirical and explanation-based learning. Knowledge Acquisition, 3, 157–173.

Pazzani, M., Brunk, C., & Silverstein, G. (1991). A knowledge-intensive approach to learning relational concepts. Proceedings of the Eighth International Workshop on Machine Learning (pp. 432–436). Evanston, IL: Morgan Kaufmann.

Pazzani, M., & Kibler, D. (1992). The role of prior knowledge in inductive learning. Machine Learning.

Pazzani. M., (1992). When Prior Knowledge Hinders Learning. AAAI workshop on constraining learning with prior knowledge. San Jose.

Quinlan, J.R., (1990). Learning logical definitions from relations. Machine Learning, 5, 239–266.

Rabinowitz, H., Flamholz, J., Wolin, E.,& Euchner, J. (1991). Nynex Max: A telephone trouble screening expert system. In R. Smith & C. Scott (Eds.) Innovative applications of artificial intelligence, 3, 213–230.

Richards, B. (1992). An Operator-Based Approach to First-Order Theory Revision. Ph.D. Thesis. University of Texas, Austin.

Richards, B. & Mooney, R. (1991). First-Order Theory Revision. Proceedings of the Eight International Workshop on Machine Learning (pp. 447–451). Evanston, IL: Morgan Kaufmann.

Segre, A. (1987). On the operationality/generality trade-off in explanation-based learning. Proceedings of the Tenth International Joint Conference on Artificial Intelligence (pp. 242–248). Milan, Italy: Morgan Kaufmann.

N94-34045

Reinforcement Learning in Scheduling

Tom G. Dietterich* DoKyeong Ok Prasad Tadepalli
Wei Zhang

Department of Computer Science
Oregon State University
Corvallis, Oregon 97331-3202

Abstract

The goal of this research is to apply reinforcement learning methods to real-world problems like scheduling. In this preliminary paper, we show that learning to solve scheduling problems such as the Space Shuttle Payload Processing and the Automatic Guided Vehicle (AGV) scheduling can be usefully studied in the reinforcement learning framework. We discuss some of the special challenges posed by the scheduling domain to these methods and propose some possible solutions we plan to implement.

1 Introduction

Scheduling is a well-studied problem and there are a variety of scheduling problems [BAKER74, FOX87, SADEH91]. Unfortunately, almost all of the nontrivial versions of the scheduling problem are at least NP-hard [GAREY79]. However, since the scheduling problems that occur in the real world may not be entirely arbitrary, there is reason to believe that there exists some structure or regularity which may not be directly apparent. The goal of this research is to build learning systems that are capable of discovering such hidden regularities in the environment of the scheduler and exploit them to efficiently build reasonably good schedules.

We are currently focusing on two specific scheduling problems. One is the problem of Space Shuttle Payload Processing [ZWEBEN92]. The goal in this domain is to schedule a set of tasks so that they finish before their respective deadlines while obeying a given set of precedence and resource constraints. The second application domain is Automatic Guided Vehicle (AGV) scheduling. The problem here is to make an on-line assignment of transportation tasks to AGVs such that the average expected cost of transport over the long run is minimized. The transportation tasks are randomly generated, and need to be serviced on-line and hence the system cannot plan the order of task execution in advance.

These two application domains are somewhat different in that the first one emphasizes the problem of efficiently searching the scheduling space, and the second one emphasizes optimal real-time decision making in a stochastic environment. Nevertheless, the same basic approach of reinforcement learning is applicable in both cases.

Reinforcement learning is an unsupervised learning method where an agent learns from the feedback or 'reinforcement' provided by the environment as a result of its actions. The reinforcement can be positive (reward) or negative (punishment) and can be delayed in time from the action that is responsible for it. The goal of the learner is to learn to behave in such a way that maximizes its total expected reward.

*The first and the last authors are supported by a grant from NASA, number 1293.

It is easy to see the correspondence between the AGV scheduling and reinforcement learning. Assume that the AGV gets several requests to transport objects from one place to another. The AGV might choose to serve these requests in some order. Whenever a request is served, the AGV gets a reinforcement which might be inversely proportional to the time delay with which the request is served. The problem then is to learn an optimal policy (a mapping from states of the world to AGV actions) that results in the maximum expected reward rate.

Although less straightforward, it is also possible to frame the space shuttle scheduling problem in the reinforcement learning context. In this context, the scheduler takes abstract scheduling actions such as moving tasks from one time slot to another, and from one machine to another. It gets a negative reinforcement proportional to the cost of the final conflict-free schedule. If the scheduler is able to correctly predict the cost of the final conflict-free schedule from intermediate states, then the search complexity can be reduced by choosing the scheduling action that will minimize the final cost of the schedule. The problem thus reduces to learning to predict the final cost of schedule from intermediate states with scheduling conflicts.

Many of the reinforcement learning methods are based on on-line versions of dynamic programming. In dynamic programming, the cost or value of a state is computed by backing up the costs of the succeeding states. Instead of computing the value of a state once and for all from the values of all its neighbors, in reinforcement learning, the value of a state is incrementally updated, as and when its neighbor's values are updated. Both the on-line and off-line versions of dynamic programming store a table of state-value pairs. The on-line version uses this table to choose an action that minimizes the expected cost of the final state.

One of the problems with the table-based reinforcement learning techniques like Q-learning is their space complexity [WATKINS92]. In the worst case, they need tables as big as the entire state space and some more, which is unrealistic in nothing but the trivial of domains. This also translates to very slow convergence of learning, because most states in the state space have to be updated many times before the learner's actions converge to optimal performance.

Reinforcement learning researchers use supervised learning methods to store the tables compactly and to converge quickly to the correct policy. For example, the table of state-value pairs can be used to train a neural net which can then predict the values of states that have not been stored. Similarly we can also consider storing the state-value pairs without generalizing them and use approaches like the nearest neighbor algorithms to predict the values for the unseen states by interpolating between the stored states which are nearest to the unseen state. Another approach would be to approximate the state-value table by a set of peiecewise polynomial functions using methods like spline interpolation. Such "structural generalization" methods give rise to a compact storage of states as well as a quicker convergence when the function they are trying to approximate suits their structure.

One of the problems in reinforcement learning is trading off exploration with optimal decision making. Exploration facilitates learning new knowledge while optimal decision making exploits old knowledge. However, most widely studied exploration strategies are random. We plan to investigate more sophisticated strategies that explore "near miss" states. These strategies seem effective for learning piece-wise polynomial functions with many locally irrelevant features.

The rest of the paper is organized as follows. The next section introduces reinforcement learning. Section 3 introduces the NASA space shuttle payload processing domain and puts it in the framework of reinforcement learning. Section 4 does the same for the AGV scheduling domain. Section 5 discusses the structural generalization problem in reinforcement learning and proposes a number of solutions that we plan to implement. Section 6 discusses some of the other challenges that the scheduling domain offers to reinforcement learning, and some proposed solutions. The last section concludes with a summary.

2 Reinforcement Learning

Reinforcement learning is best suited to a class of stochastic optimal control problems called Markovian decision problems [BARTO93].

The reinforcement learning problem can be described by a 4-tuple $\langle S, A, p, C \rangle$. S is a finite set of states and A is the set of actions. $p_{i,j}(a)$ is the probability for every state pair (i, j) and action $a \in A$, that executing action a in state i results in state j. The Markovian assumption means that this probability is independent of the states before i. $c_t = C(i_t, a_t)$ denotes the immediate cost or reward of executing an action a_t in a state i_t at time t. In some versions of the problem, when the horizon is not finite, the cost or reward is discounted by multiplying it with a discount factor γ^t. where

A policy μ is a mapping from the state i_t at time t to a recommended action in A. f^μ denotes the expected value of the infinite-horizon discounted cost, given by:

$$f^\mu(i_t) = E_\mu \left[\Sigma_{j=0}^{\infty} \gamma^{t+j} c_{t+j} \right]$$

where E_μ is the expectation assuming that the controller always follows policy μ. An optimal policy μ^* is one that minimizes f^μ, and f^* is its expected discounted cost.

Knowing f^* would allow one to construct μ^*, because, by the results of dynamic programming, any policy which is greedy (chooses the action that results in the least cost state) with respect to f^* is optimal.

Various versions of dynamic programming (DP) compute f^* by backing up costs from the last state to previous states in different orders [BARTO93]. Reinforcement learning methods use on-line versions of dynamic programming. Some of these methods, including Q-learning, do not assume that the transition probability matrix p is known. Q-learning eliminates the need for separately learning the p matrix, by learning a so called Q-function from state action pairs to the expected discounted costs of taking that action in that state. In particular, if i_t is the current state and a_t is the action taken, then

$$Q(i_t, a_t) = \Sigma_{j=0}^{\infty} \gamma^j c_{t+j}$$

Since the value of a state $V(i)$ is the value of the best action in that state, we can write

$$V(i_t) = \min_a Q(i_t, a)$$

and, it follows that

$$Q(i_t, a_t) = c_t + \gamma V(i_{t+1})$$

In Q-learning, the Q values of the final "absorbing states" can be immediately calculated, which are backed up from last to first for the states the system has passed through. More formally, the Q-function is computed in stages as follows. If S_k is the sequence of observed states and actions $(s_1, a_1, s_2, \ldots)$ at stage k, and α is the learning rate, Q_k is updated for states in S_k from last to first as follows:

$$Q_k(i_t, a_t) = (1 - \alpha) Q_{k-1}(i_t, a_t) + \alpha(c_t + \gamma V_{k-1}(i_{t+1})),$$
$$\text{where, } V_k(i_t) = \min_a Q_k(i_t, a)$$

The theoretical results show that if the system explores every state infinitely often, then it eventually converges to the optimal Q values for all the state action pairs.

3 The Space Shuttle Payload Processing

The NASA Space Shuttle Payload Processing domain is an example of Job Shop Scheduling problem [ZWEBEN92]. Each 'mission' consists of a set of payload/carrier pairs, and a launch date. Each carrier requires a distinct set of tasks to prepare and process the payloads for a mission. The tasks are constrained by precedence and resource relationships. The resources are grouped into resource pools. The goal is to schedule all the tasks needed to load the carriers onto the orbiter, avoiding the resource contention problems, satisfying all precedence constraints while minimizing the total expected length of the schedule. More details can be found in [ZHANG93].

The Space Shuttle Payload Processing problem can be viewed as a state space search problem, where states are partial schedules with possible conflicts, and operators move from state to state by moving tasks. The problem is to find a conflict-free schedule of minimum length. Unlike in some other domains, there is a lot of flexibility in defining the operators in the scheduling domain. Individual tasks can be moved by a constant amount, or by an amount that depends on the availability of resources and the schedule of other tasks, or groups of tasks can be rescheduled using a single operator. One could also consider a hierarchy of abstract to more primitive operators. We plan to experiment with all these different options.

The search control knowledge for scheduling is expressed as an evaluation function that estimates the discounted final cost of the schedule reachable from the current state. In reinforcement learning methods like $TD(\lambda)$, this amounts to an estimate of f^* [SUTTON88]. Q-learning estimates it from the state that results by applying a scheduling action to the current state.

If the evaluation function is accurate, then it can be used to select the action that leads to the state with the least cost without search. When the evaluation function is only approximate, as is likely to be the case in complex domains like scheduling, it can be combined with look ahead search, as done in 2-person games like chess, to exploit the benefits of both knowledge and search.

4 The AGV Scheduling Problem

Automatic Guided Vehicles or AGVs are increasingly being used in manufacturing plants to cut down the cost of human labor in transporting materials from one place to another. Optimal scheduling of AGVs is a non-trivial task. In general, there can be multiple AGVs, with some routing constraints, e.g., two AGVs cannot be on the same route fragment going in opposite directions. The AGVs might also have capacity constraints such as the total load and volume they can carry, and the total time they can work without recharging.

The transportation requests are stochastic and hence cannot be planned for in advance. The AGV gets a reward whenever it successfully serves a request. The behavior of the AGV is random in the beginning, but as it accumulates knowledge of the request patterns and the transportation costs involved, we expect it to perform better in the sense of serving more requests in a given time. In the reinforcement learning context, this corresponds to maximizing the average reward per unit time rather than maximizing the discounted reward. We can also associate a non-uniform reward structure with the requests and give more reward for serving some requests and not the others.

A learning AGV is very attractive in a manufacturing plant because the scheduling environment is constantly changing and it is hard to manually optimize the scheduling algorithm to each changed situation. A learning AGV would automatically adapt itself changes in its environment, be they are added machines, changes in the AGV routes, or changes in the request rates and patterns.

Once again, we treat the AGV scheduling as a state space search, and treat the status of various requests and the AGV as a state. The best action to take at any time depends on the current state. The

optimal policy can be learned using methods like Q-learning.

5 Structural Generalization in Reinforcement Learning

One of the major issues in reinforcement learning is that of structural generalization. This can also be seen as choosing a representation for the evaluation function. An extreme representation is as a table of mappings from state descriptions to their evaluations. This amounts to no structural generalization at all, since no two state descriptions will have the same entry in the table. Representing it as a table of mappings is infeasible in many scheduling domains due to their size and slow convergence. One of the requirements is also that real-valued functions should be representable.

One of the popular representations of evaluation functions is neural nets. Recently a program that used TD(λ) in combination with neural nets to represent its evaluation function to learn to play Backgammon reached grand-master level performance and is ranked as the best Backgammon program in the world [TESAURO92]. The success of neural net learning crucially depends on being able to find good encodings for states and operators.

Another reasonable representation is piece-wise polynomial functions. These functions can be learned by nearest neighbor algorithms. In the extreme version of these algorithms, no generalization is necessary. Each example is stored as an input-output pair. To predict the output for an unseen example, its nearest (using some distance metric like the Euclideanm distance) K neighbors are examined and the output is calculated to be the weighted sum of their outputs. One of the disadvantages of this approach is that it needs a lot of memory to store all the examples, and hence the name "memory-based" [MOORE93]. An optimization that is usually done is to store an example only when the current set of examples does not make a correct prediction for this example.

6 Research Issues

The scheduling domain offers some interesting challenges to reinforcement learning.

One of the complexities of the scheduling problems like the space shuttle scheduling is that they have a large number of applicable operators at any state leading to a search space with a large branching factor. When the branching factor is large, it is not realistic to choose the best action by trying each possible action and comparing the results, as the standard greedy policy adopted in reinforcement learning methods usually does. In this case, we can use a random sample greedy strategy which chooses the best action from among a randomly sampled subset of all possible actions. We also plan to experiment with methods such as simulated annealing and gradient descent search, which do not involve testing each possible next state.

There are also usually a large number of irrelevant features in the state description of the scheduling problems. The presence of irrelevant features makes it difficult to generalize correctly. For example, in the nearest neighbor approach, the irrelevant features distort the distances between examples so that examples which are close in relevant features appear distant with irrelevant features and vice versa. Since the number of examples needed to converge in this approach varies exponentially with the number of features, a large number of irrelevant features works against this approach [AHA91]. Adjustable feature weights has been proposed as a solution for this problem [AHA91]. While this method eliminates globally irrelevant features, one problem with it is that it does not take local irrelevancy into account.

One way to determine local irrelevancy is through intelligent exploration. Most reinforcement learning methods use random exploration to gain new knowledge about their search spaces. However, if one knows that the evaluation function has certain structure, say that it is representable as a piece-wise polynomial

function with many locally irrelevant features, then it may be possible to explore this search space more intelligently. For example, it may be possible to determine locally irrelevant features by generating "near miss" examples, which are examples which differ from a base example by exactly one feature in a small amount, but change the value of the function. The existence of a near miss example in a feature shows the local relevance of that feature. Generating near miss examples and determining the values of the evaluation function at these examples add the ability of intelligent exploration to reinforcement learning.

7 Conclusions

In this paper, we suggested that reinforcement learning can be usefully employed in scheduling domains to learn search control knowledge as well as to learn to do optimal real-time scheduling. The work reported here is preliminary and much remains to be done. We plan to implement the ideas reported in this paper, test them and report the results in the near future.

References

[AHA91] Aha, D. W., Kibler, D., and Albert, M. K. Instance-based learning algorithms. *Machine Learning, 6* (1), 37–66. 1991.

[BAKER74] Baker, K.R., Introduction to Sequencing and Scheduling, John Wiley & Sons, Inc., 1974.

[BARTO93] Barto, A. G., Bradtke, S. J., and Singh S. P. Learning to Act using Real-Time Dynamic Programming. To appear in *AI Journal* special issue on Computational Theories of Interaction and Agency.

[FOX87] Fox, M. S. *Constraint-Directed Search: A Case Study of Job-Shop Scheduling.* Morgan Kaufmann, Los Altos, CA, 1987.

[GAREY79] Garey M.R., and Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman and Company, New York, 1979.

[MOORE93] Moore, A.W. and Atkeson, C. G. Memory-based Reinforcement Learning: Converging with less data and less real time. (to appear).

[SADEH91] Sadeh, N. Look-ahead Techniques for Micro-opportunistic Job Shop Scheduling. Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, available as CMU-CS-91-102, 1991.

[SUTTON88] Sutton, R. S. Learning to Predict by the Methods of Temporal Difference. In *Machine Learning*, 3, 9-44, 1988.

[TESAURO92] Tesauro, G. Temporal Difference Learning of Backgammon Strategy. In *Proceedings of Machine Learning Conference*, 1992.

[WATKINS92] Watkins, C. J. C. H. and Dayan P. Q-learning. *Machine Learning*, 8:279-292, 1992.

[ZHANG93] Zhang, W. A Study of Reinforcement Learning for Job-shop Scheduling. Ph.D. Thesis proposal, Oregon State University, Corvallis, OR, 1993.

[ZWEBEN92] Zweben, M., Davis, E. Daun, B., Drascher, E., Deale, M., and Eskey, M. Learning to improve constraint-based scheduling. *Artificial Intelligence*, 58:271-296, 1992.

Constraint-Based Integration of Planning and Scheduling for Space-Based Observatory Management*

Nicola Muscettola and Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
phone: (412) 268-8811
net: sfs@isl1.ri.cmu.edu

Introduction

The generation of executable schedules for space-based observatories is a challenging class of problems for the planning and scheduling community. Existing and planned space-based observatories vary in structure and nature, from very complex and general purpose, like the Hubble Space Telescope (HST), to small and targeted to a specific scientific program, like the Submillimeter Wave Astronomy Satellite (SWAS). However, they all share several classes of operating constraints including periodic loss of target visibility, and limited on-board resources like battery charge and data storage.

The complexity of these problems stems from two sources. First, the execution of astronomy observation programs requires the solution of a classical scheduling problem: objectives relating to overall system performance must be optimized (e.g., maximization of return of science data) while satisfying a diverse set of constraints. These constraints relate to both the observation programs to be executed (e.g., precedence and temporal separation among observations) and observatory capacity limitations (e.g., observations requiring different targets cannot be executed simultaneously). Second, a safe mission requires the detailed description of all transitions and intermediate states that support the achievement of observing goals. Such description must guarantee consistency with respect to the detailed dynamics of the observatory; this constitutes a classical planning problem.

Another characteristic of the problem is its large scale. The size of the pool of observations to be performed on a yearly horizon can range from thousands to even tens of thousands. Large observatories can consist of several tens of interacting system components with complex interacting dynamics. To effectively deal with problems of this size, it is essential to employ problem and model decomposition techniques. In certain cases, this requires an ability to represent and exploit the available static structure of the problem (e.g., interacting system components). In other cases an explicit structure is not immediately evident (e.g., interaction among large numbers of temporal and capacity constraints); therefore the problem solver should be able to dynamically focus on different parts of the problem, exploiting the structure that emerges during the problem solving process itself.

In this paper, we report on our progress toward the development of effective, practical solutions to space-based observatory scheduling problems within the HSTS scheduling framework. HSTS was developed and originally applied in the context of the HST short-term observation scheduling problem. Our work was motivated by the limitations of the current solution and, more generally, by the insufficiency of classical planning and scheduling approaches in this problem context. HSTS has subsequently been used to develop improved heuristic solution techniques in related scheduling domains, and is currently being applied to develop a scheduling tool for the upcoming SWAS mission. We first summarize the salient architectural characteristics of HSTS and their relationship to previous scheduling and AI planning research. Then, we describe some key problem decomposition techniques underlying our integrated planning and scheduling approach to the HST problem; research results indicate that that these technique provide leverage in solving space-based observatory scheduling problems. Finally, we summarize more recently developed constraint-posting scheduling procedures and our current SWAS application focus.

Planning and Scheduling for Space-Based Observatories

The management of the scientific operations of the Hubble Space Telescope is a formidable task. Its solution is the unique concern of an entire organization, the Space Telescope Science Institute (STScI). The work of several hundred people is supported by several software tools, organized in the Science Operations Ground System (SOGS). At the heart of SOGS is a FORTRAN-based software scheduling system, SPSS. The original task of SPSS was to take astronomer viewing programs for a yearly period as input and produce executable spacecraft instructions as output, with minimal intervention from human operators. SPSS has had a somewhat checkered history [Wal89], due in part to the complexity of the scheduling problem and in part to the difficulty of developing a solution via traditional software engineering practices and conventional

*The work reported in this paper has be supported in part by the National Aeronautics and Space Administration, under contract NCC 2-531, and by the Advanced Research Projects Agency under contract F30602-90-C-0119.

programming languages. To confront SPSS's computational problems, STScI has developed a separate, knowledge-based tool for long term scheduling called SPIKE [Joh90]. SPIKE accepts programs approved for execution in the current year and partitions observations into weekly time buckets. Each bucket constitutes a smaller, more tractable, short-term scheduling problem. Detailed weekly schedules are then generated through the efforts of a sizable group of operations astronomers, who interactively utilize SPSS to place observations on the time line.

In the HSTS project we have addressed the short term problem in the HST domain, i.e., the problem of efficiently generating detailed schedules that account for the major operational constraints of the telescope and for the domain's optimization objectives. The basic assumption is to treat resource allocation (scheduling) and auxiliary task expansion (planning) as complementary aspects of a more general process of constructing behaviors of a dynamical system. [Mus90].

Two basic mechanisms provide the basis of the HSTS approach:

1. a *domain description language* for modeling the structure and dynamics of the physical system at multiple levels of abstraction.
2. a *temporal data base* for representing possible evolutions of the state of the system over time (i.e. schedules).

In HSTS the natural approach to problem solving is by iterative posting of constraints, extracted either from the external goals or from the description of the system dynamics. Consistency is tested through constraint propagation. For more details, see [MSCD92, Mus93b].

Three key aspects distinguish HSTS from other approaches:

1. the state of the modeled system is explicitly decomposed into a finite set of "state variables" evolving over continuous time. This enables the specification of algorithms exploiting problem decomposability, and provides the necessary structure for optimizing resource utilization.
2. the temporal data base permits flexibility along both temporal and state value dimensions. The time of occurrence of each event does not need to be fixed but can float according to the temporal constraints imposed on the event by the process of goal expansion. This flexibility contributes directly to scheduling efficiency. Since overcommitment can be avoided, there is a lower possibility of backtracking.
3. the constraint posting paradigm accommodates a range of problem solving strategies (e.g. forward simulation, back chaining, etc.). This allows the development of algorithms that opportunistically exploit problem structure to consistently direct problem solving toward the most critical tradeoffs.

The importance of integrating these three features within a single framework can be appreciated by considering the limitations of other approaches that address them separately or partially.

In planning, most Artificial Intelligence research adopts the classical representational assumption proposed by the STRIPS planning system [FHN72]. In this view action is essentially an instantaneous transition between two world states of indeterminate durations. The structural complexity of a state description is not limited, but the devices provided for its description are completely unstructured, such as complete first order theories or lists of predicates. Some frameworks [Wil88, CT91] have demonstrated the ability to address practical planning problems. However, the classical assumption lacks balance between generality and structure; this is a major obstacle in extending classical planning into integrated planning and scheduling. Past research has attempted partial extensions in several important directions: processes evolving over continuous [AK83] and metric time [Ver83, DFM88], parallelism [Lan88], and external events [For89]. However, no comprehensive view has yet been proposed to address the integration problem.

Classical scheduling research has always exploited much stronger structuring assumptions [Bak74]. Domains are decomposed into a set of resources whose states evolve over continuous time. This facilitates the explicit representation of resource utilization over extended periods of time. Several current scheduling systems exploit reasoning over such representations [FS84, SOM+90, Sad91, MJPL92, ZG90, BC91]. Empirical studies have demonstrated the superiority of this approach [OS88, Sad91] with respect to dispatching scheduling [PI77], where decision making focuses only on the immediate future. However, the scheduling view of the world also has very strong limitations. No information is kept about a resource state beyond its availability. Additional state information (e.g., in which direction the observatory is pointing at a given time) is crucial to maintain causal justifications and to dynamically expand support activities during problem solving.

Issues in Integrating Planning and Scheduling

Use of Abstraction

The use of abstract models has long been exploited as a device for managing the combinatorics of planning and scheduling. In HSTS models are expressed in terms of interacting state variables representing different components of the system (in our case, the space-based observatory) and of its operating environment (e.g., celestial objects to be observed). An abstract model can summarize system dynamics with state variables that aggregate several structural components; alternatively abstract models can selectively simplify system dynamics by omitting one or more component state variables. Given the structure of space-based observatory scheduling problems, an abstract model provides a natural basis for isolating overall optimization concerns. This provides global guidance in the development of detailed, executable schedules.

In the case of HST, a two-level model has proved sufficient. At the abstract level, telescope dynamics is summarized in terms of a single state variable, indicat-

ing, at any point in time, whether the telescope (as a whole) is taking a picture, undergoing reconfiguration, or sitting idle. At this level reconfiguration transitions have duration constraints that estimate the time required by the actual reconfiguration activities implied by the detailed model (e.g., instrument warmup and cool-down, data communication, telescope repointing). Execution of an observation at the abstract level requires only satisfaction of this abstract reconfiguration constraint, target visibility constraints, and any user specified temporal constraints. Thus, the description at the abstract level looks much like a classic scheduling problem: a set of user requests that must be sequenced on a single resource subject to specified constraints and allocation objectives.

Planning on a fully detailed model ensures the viability of any sequencing decisions made at the abstract level. This corresponds to generating and coordinating required supporting system activities. The degree of coupling between reasoning at different levels depends in large part on the accuracy of the abstraction. In the case of HST, decision-making at abstract levels is tightly coupled; each time a new observation is inserted into the sequence at the abstract level, control passes to the detailed level to develop detailed segments of system behavior necessary to achieve the new goal. Given the imprecision in the abstract model, goals posted for detailed planning cannot be rigidly constrained; instead only preferences are specified (e.g., "execute as soon as possible after obs1"). The results of each detailed planning stage are propagated at the abstract level to provide more precise constraints for subsequent abstract level decision-making.

Model Decomposability and Incremental Scaling

To approach large problems it is typically necessary to decompose them into smaller sub-problems, solve each sub-problem separately, and then assemble the sub-solutions. We can judge how the problem solving framework supports modularity and scalability if it displays the following two features:

- the search procedure for the entire problem can be assembled by combining heuristics independently developed for each sub-problem, with little or no modification of the heuristics;
- the computational effort needed to solve the complete problem does not increase with respect to the sum of the efforts needed to solve each component sub-problem.

We conducted an experiment with three increasingly complex and realistic models of the HST, in order to evaluate the support that HSTS provides towards the realization of these features (issues relating to optimization of the overall mission performance criteria will be discussed in the following sections).

We identify the three models as SMALL, MEDIUM, and LARGE. All share the same abstract level representation. At the detail level the three models include state variables for different telescope functionalities. The SMALL model has a state variable for the visibility

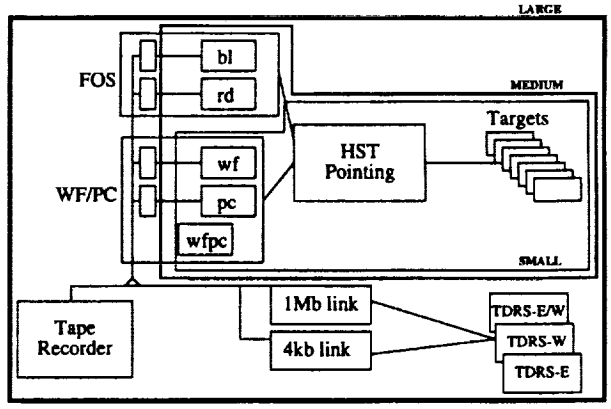


Figure 1: The SMALL, MEDIUM and LARGE HST models.

of each target of interest, a state variable for the pointing state of the telescope, and three state variables to describe a single instrument, the Wide Field/Planetary Camera (WFPC). The MEDIUM model includes SMALL and two state variables for an additional instrument, the Faint Object Spectrograph (FOS). Finally, the LARGE model extends MEDIUM with eight state variables accounting for data communication. The LARGE model is representative of the major operating constraints of the domain. Figure 1 shows the relations among the various models.

For each model we use the same pattern of interaction between problem solving at the abstract and at the detail level. At the abstract level observations are selected and dispatched using a greedy heuristic to minimize expected reconfiguration time. The last dispatched observation is refined into the corresponding detail level problem; then control is passed to planning/scheduling at the detail level. This cycle is repeated until the abstract level sequence is complete.

The detail planner/scheduler for SMALL is driven by heuristics which deal with the interactions among its system components. A first group ensures the correct synchronization of the WFPC components; one of them, for example, states that, when planning to turn on the WF detector, preference should be given to synchronization with a PC behavior segment already constrained to be off. A second group deals with the pointing of HST; for example, one of them selects an appropriate target visibility window to execute the locking operation. A final group manages the interaction between the state of WFPC and target pointing; an example from this group states a preference to observe while the telescope is already scheduled to point at the required target. To solve problems in the context of MEDIUM, additional heuristics must deal with the interactions within FOS components, between FOS and HST pointing state, and between FOS and WFPC. However, the nature of these additional interactions is very similar to those found in SMALL. Consequently, it is sufficient to extend the domain of applicability of SMALL's heuristics to obtain a complete set of heuristics for MEDIUM. For example, the heuristic excluding

Model	SMALL	MEDIUM	LARGE
State Variables	4	6	13
Tokens	587	604	843
Time Points	588	605	716
Temporal Constraints	1296	1328	1474
CPU Time / Observation	11.62	12.25	21.74
CPU Time / Compatibility	0.29	0.29	0.33
Total CPU time	9:41.00	10:11.50	18:07.00
Total Elapsed Time	1:08:36.00	1:13:16.00	2:34:07.00
Schedule Horizon	41:37:20.00	54:25:46.00	52:44:41.00

Table 1: Performance results. The times are reported in hours, minutes, seconds, and fractions of second

WF and PC from being in operation simultaneously can be easily modified to ensure the same condition among the two FOS detectors. Finally, for LARGE we include the heuristics used in MEDIUM with no change, plus heuristics that address data communication and interaction among instruments and data communication; an example of these prevents scheduling an observation on an instrument if data from the previous observation has not yet been read out of its data buffer. By making evident the decomposition in modules and the structural similarities among different sub-models, HSTS made possible the reuse of heuristics and their extension from one model to another. We therefore claim that HSTS displays the first feature of a modular and scalable planning/scheduling framework.

To determine the relationship between model size and computational effort, we ran a test problem in each of the SMALL, MEDIUM, and LARGE models. Each test problem consisted of a set of 50 observation programs; each program consisted of a single observation with no user-imposed time constraints. The experiments were run on a TI Explorer II+ with 16 Mbytes of RAM memory.

As required by the second feature of a scalable framework, the results in table 1 indicate that the computational effort is indeed additive. In the table, the measure of model size (number of state variables) excludes visibilities for targets and communication satellites, since these can be considered as given data. The temporal constraints are links between two time points that lie on different state variables; the number of these links gives an indication of the amount of synchronization needed to coordinate the evolution of the state variables in the schedule.

Since the detail level heuristics exploit the modularity of the model and the locality of interactions, the average CPU time (excluding garbage collection) spent posting an elementary temporal relation constraint (compatibility) remains relatively stable. In particular, given that the nature of the constraints included in SMALL and MEDIUM is very similar, the time is identical in the two cases. The total elapsed time to generate a schedule in the case of LARGE is an acceptable fraction of the time horizon covered by the schedule during execution. Even if this implementation is far from optimal, it nonetheless shows the practicality of the framework for the actual HST operating environment.

Exploiting Opportunism to Generate Good Solutions

In space-based observatory scheduling a critical tradeoff is between: (1) maximizing the time spent collecting science data; (2) satisfying absolute temporal constraints associated with specific user requests. The scheduling problem is typically over-subscribed, i.e., it will generally not be possible to accommodate all user requests in the current short term horizon, and some must necessarily be rejected. A lost opportunity corresponds to the rejection of a request whose user-imposed time windows fall inside the current scheduling horizon. Observation requests without such execution constraints are not lost because the scheduler may reattempt to honor them in subsequent scheduling episodes.

The experiment described in the previous section uses a dispatch-based strategy for sequence development. Simulating forward in time at the abstract level, the strategy repeatedly added to the end of the current sequence the candidate observation estimated to incur the minimum amount of wait time (due to HST reconfiguration and target visibility constraints). This heuristic strategy, termed "nearest neighbor with look-ahead" (NNLA), attends directly to the first global objective of maximizing the time spent collecting science data.

However, forward simulation does not sufficiently address the second global objective: the minimization of rejections of absolutely constrained requests. A request's window of opportunity may be gone by the time the scheduler rates the request as the choice with minimum dead time. Coupling forward simulation with look-ahead search (i.e. evaluation of possible "next sequences" and potential rejections) can provide protection against unnecessary request rejection. However this approach has limited effectiveness because of combinatorics. A second sequencing strategy directly attends to the minimization of the number of rejected requests with absolute constraints: "most temporally constrained first" (MCF). MCF's computational complexity is comparable to that of NNLA. Under the MCF scheme, the sequence is built by repeatedly selecting the pending request with the tightest execution bounds and inserting it in the schedule. Unlike NNLA this strategy does not build a sequence with a simulation-based approach. Honoring the temporal constraints of the selected requests will create availability "holes" over the scheduling horizon. Incidentally, the MCF strategy is quite close to the algorithm currently employed in the operational system at STScI.

As is the case with the NNLA strategy, one objective is also emphasized at the expense of the other within the MCF strategy. The availability holes opened by MCF can result in considerable telescope idle time and therefore a sequence insertion heuristic should seek to minimize such dead time. However, its effectiveness is largely determined by the specific characteristics and distribution over the horizon of the initially placed requests.

Both NNLA and MCF manage combinatorics by

Sequencing Strategy	Pctg. Constrained Goals Scheduled	Pctg. Telescope Utilization
NNLA	72	21.59
MCF	93	17.20
MCF/NNLA	93	20.54

Table 2: Comparative Performance of NNLA, MCF and MCF/NNLA

making specific problem decomposition assumptions and localizing search according to them. NNLA assumes an event based decomposition (considering only the immediate future) while MCF assumes that the problem is decomposable by degree of temporal constrainedness. Previous research in constraint-based scheduling [SOM+90] has indicated the leverage of dynamic problem decomposition and selective use of local scheduling perspectives. In our case, we can also evaluate problem structure to select the appropriate strategy between NNLA and MCF at any point during sequence development. In particular we estimate the current variance of the number of feasible start times remaining for individual unscheduled requests. If the variance is high, there is an indication that some remaining requests might be much more constrained than others; this prompts the use of MCF to emphasize placement of tightly constrained goals. If the variance is low, there is an indication that all pending requests have similar temporal flexibility; the emphasis can switch to minimizing dead time within current availability "holes" using NNLA.

To test this multi-perspective approach, we solved a set of short-term (i.e., daily) scheduling problems using three separate strategies: NNLA, MCF and the composite strategy just described (referred to as MCF/NNLA). The results are given in Table 2. They confirm our expectations as to the limitations of both NNLA and MCF. We can also see that MCF/NNLA produces schedules that more effectively balance the two competing objectives. Further details on the sequencing strategies and the experimental may be found in [SP92].

These results should be viewed as demonstrative and we are not advocating MCF/NNLA as a final solution. We can profitably exploit other aspects of the current problem structure and employ other decomposition perspectives. For example, the distribution of goals over the horizon implied by imposed temporal constraints has proved to be a crucial guideline in other scheduling contexts [SOM+90, Sad91], and we are currently investigating the use of analogous look-ahead analysis techniques within the problem solving framework provided by HSTS (see below). There are also additional scheduling criteria and preferences (e.g., priorities) in space-based observatory domains that are currently not accounted for.

Constraint Posting Scheduling

Most constraint-based scheduling research addresses the problem of finding a single, consistent assignment of start times for each activity [BC91, Joh90, KY89, MJPL92, SOM+90, Sad91, ZG90]. HSTS, in contrast, advocates a problem formulation more akin to least-commitment planning frameworks. The problem is most naturally treated as one of posting sufficient additional precedence constraints between pairs of activities so as to ensure feasibility with respect to time and capacity constraints. Solutions generated in this way typically represent a set of feasible schedules (i.e., the sets of activity start times that remain consistent with posted sequencing constraints), as opposed to a single assignment of start times.

While HSTS does not prohibit the use of "fixed time" scheduling techniques, there are several potential advantages to a solution approach that retains solution flexibility as problem constraints permit. A flexible schedule provides a measure of robustness against uncertainty during schedule execution. The determination of actual start times can be delayed until execution and solution revision can be minimized. A constraint posting approach can also provide a more convenient search space in which to operate during schedule development. Alternatives are not unnecessarily pruned by (over) committing on specific start times. When the need for schedule revision becomes apparent, modifications can often be made much more directly and efficiently through simple adjustment of posted constraints. Our recent research has developed and evaluated two novel algorithms for generating schedules via constraint posting that demonstrate the previous potential advantages: Conflict Partition Scheduling (CPS)[Mus92] and Precedence Constraint Posting Scheduling (PCP)[SC93].

The CPS procedure builds on previous research into techniques for estimating resource contention[MS87]. Great emphasis is put in the recognition of resource capacity bottlenecks - time periods with high predicted contention among activities for the same resource capacity. In CPS capacity bottleneck are detected through use of a stochastic simulation technique. After identifying resource capacity bottlenecks, CPS acts to lessen the level of contention by posting ordering constraints among the activities competing for capacity at the most severe bottleneck. The iterative process continues until no capacity conflict remains; at this point a final schedule has been determined. If the process reaches an infeasible solution state, the search is simply restarted. CPS has been experimentally tested on a set of benchmark constraint satisfaction scheduling problems. The performance analysis demonstrated superior problem solving performance to two currently dominant "fixed-times" scheduling approaches - micro-opportunistic scheduling [Sad91] and min-conflict iterative repair [MJPL92]. The reader is referred to [Mus92] for details. More recent work has aimed at the evaluation of different alternative CPS configurations (e.g., micro vs macro decision making, focused on capacity conflicts vs randomly focused) to establish the relative importance of different steps of the procedure and the corresponding performance trade-offs [Mus93a].

The PCP procedure combines two techniques: dom-

inance conditions for incremental pruning of the set of feasible sequencing alternatives [EV76] and a simple look-ahead analysis of the temporal flexibility associated with different sequencing decisions. At each step of the search, a measure of "residual temporal slack" is computed for each sequencing decision that remains to be made. PCP choses the decision with the smallest residual slack as the most critical, and posts a precedence constraint in the direction that retains the most flexibility. Posting a constraint might leave other sequencing decisions with only a single feasible ordering; these unconditional decisions are taken by posting the implied precedence before recomputing estimates of residual slack. Unlike CPS, which posts constraints only until all resource contention has been resolved, PCP terminates when either all pairs of activities contending for the same resource have been sequenced, or an infeasible state has been reached. PCP has also been tested on the same suite of constraint satisfaction used in the performance analysis of CPS. PCP has shown comparable problem solving performance to other contention-based scheduling approaches at a small fraction of the computational cost[SC93].

Our current research pursues the following goals: (1) extension of both the CPS and PCP approaches to address optimization criteria; (2) investigation of complementary techniques for exploiting solution flexibility in reactive contexts; (3) evaluation of the effectiveness of these techniques in the context of space-based observatory scheduling problems.

Conclusions

In this paper, we have considered the solution of space-based observatory scheduling problems. These problems require a synthesis of the processes of resource allocation and expansion of auxiliary activities. We briefly outlined the HSTS framework and contrasted it with other scheduling and AI planning approaches. To illustrate the adequacy of the framework, we then examined its use in solving the HST short-term scheduling problem. We identified three key ingredients to the development of an effective, practical solution: flexible integration of decision-making at different levels of abstraction, use of domain structure to decompose the planning problem and facilitate incremental solution development/scaling, and opportunistic use of emergent problem structure to effectively balance conflicting scheduling objectives. The HSTS representation, temporal data base, and constraint-posting framework provide direct support for these mechanisms. Finally, we summarized more recent research aimed at further demonstration of the efficacy of constraint posting scheduling in HSTS.

We are currently utilizing HSTS to develop a scheduling tool for support of the upcoming Submillimeter Wave Astronomy Satellite (SWAS) mission. This scheduling domain requires attendance to many types of scheduling constraints similar to the the HST domain (e.g., target visibility, power). However, the SWAS scheduling problem also differs in character from the HST problem; whereas the HST problem involves synchronization of sets of observations with prespecified targets and durations, the SWAS scheduling requirement is to distribute viewing (or data integration) time among a prioritized set of desired targets. We have developed and are currently experimenting with an initial solution to the SWAS problem. This consists of a heuristic algorithm that utilizes target priority and dead time minimization criteria to create and interleave individual target observations of various durations. These results indicate the flexibility of HSTS to accommodate different types of scheduling problems. Current plans call for refinement and subsequent transfer of this solution to the SWAS mission team by the end of the year.

References

[AK83] J. Allen and J.A. Koomen. Planning using a temporal world model. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 741–747, 1983.

[Bak74] K.R. Baker. *Introduction to Sequencing and Scheduling.* John Wiley and Sons, New York, 1974.

[BC91] E. Biefeld and L. Cooper. Bottleneck identification through chronology-directed search. In *Proceedings 12th International Conference on Artificial Intelligence*, pages 218–224, Sydney, Australia, August 1991.

[CT91] K. Currie and A. Tate. O-plan: the open planning architecture. *Artificial Intelligence*, 52(1):49–86, 1991.

[DFM88] T. Dean, R.J. Firby, and D. Miller. Hierarchical planning involving deadlines, travel time, and resources. *Computational Intelligence*, 4:381–398, 1988.

[EV76] F. Roubellat Erschler, J. and J.P. Vernhes. Finding some essential characteristics of the feasible solutions for a scheduling problem. *Operations Research*, 24:772–782, 1976.

[FHN72] R.E. Fikes, P.E. Hart, and N.J. Nilsson. Learning and executing generalized robot plans. *Artificial Intelligence*, 3:251–288, 1972.

[For89] K.D. Forbus. Introducing actions into qualitative simulation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1273–8. Morgan Kaufmann, 1989.

[FS84] M.S. Fox and S.F. Smith. Isis: A knowledge-based system for factory scheduling. *Expert Systems*, 1(1):25–49, 1984.

[Joh90] M.D. Johnston. Spike: Ai scheduling for nasa's hubble space telescope. In *Proceedings of the 6th IEEE Conference on Artificial Intelligence Applications*, pages 184–190, 1990.

[KY89] N. Keng and D.Y. Yun. A planning/scheduling methodology for the constrained resource problem. In *Proceedings IJCAI-89*, Detroit, MI., Aug 1989.

[Lan88] A. Lansky. Localized event-based reasoning for multiagent domains. *Computational Intelligence*, 4:319–340, 1988.

[MJPL92] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58:361–205, 1992.

[MS87] N. Muscettola and S.F. Smith. A probabilistic framework for resource-constrained multi-agent planning. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 1063–1066. Morgan Kaufmann, 1987.

[MSCD92] N. Muscettola, S.F. Smith, A. Cesta, and D. D'Aloisi. Coordinating space telescope operations in an integrated planning and scheduling architecture. *IEEE Control Systems Magazine*, 12(1), February 1992.

[Mus90] N. Muscettola. Planning the behavior of dynamical systems. Technical Report CMU-RI-TR-90-10, The Robotics Institute, Carnegie Mellon University, 1990.

[Mus92] N. Muscettola. Scheduling by iterative partition of bottleneck conflicts. Technical report, The Robotics Institute, Carnegie Mellon University, 1992.

[Mus93a] N. Muscettola. An experimental analysis of bottleneck-centered opportunistic scheduling. Technical Report CMU-RI-TR-93-06, The Robotics Institute, Carnegie Mellon University, March 1993.

[Mus93b] N. Muscettola. Hsts: Integrating planning and scheduling. Technical Report CMU-RI-TR-93-05, The Robotics Institute, Carnegie Mellon University, March 1993.

[OS88] P.S. Ow and S.F. Smith. Viewing scheduling as an opportunistic problem solving process. In R.G. Jeroslow, editor, *Annals of Operations Research 12*. Baltzer Scientific Publishing Co., 1988.

[PI77] S.S. Panwalker and W. Iskander. A survey of scheduling rules. *Operations Research*, 25:45–61, 1977.

[Sad91] N. Sadeh. *Look-ahead Techniques for Micro-opportunistic Job Shop Scheduling*. PhD thesis, School of Computer Science, Carnegie Mellon University, March 1991.

[SC93] S.F. Smith and C. Cheng. Slack-based heuristics for constraint satisfaction scheduling. In *Proceedings AAAI-93*, Washington DC, July 1993.

[SOM+90] S.F. Smith, J.Y. Ow, P.S. Potvin, N. Muscettola, , and D. Matthys. An integrated framework for generating and revising factory schedules. *Journal of the Operational Research Society*, 41(6):539–552, 1990.

[SP92] S.F Smith and D.K. Pathak. Balancing antagonistic time and resource utilization constraints in over-subscribed scheduling problems. In *Proceedings 8th IEEE Conference on AI Applications*, March 1992.

[Ver83] S. Vere. Planning in time: Windows and durations for activities and goals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, 1983.

[Wal89] M. Waldrop. Will the hubble space telescope compute ? *Science*, 243:1437–1439, March 1989.

[Wil88] D.E. Wilkins. *Practical Planning*, volume 4. Morgan Kaufmann, 1988.

[ZG90] M. Deale Zweben, M. and R. Gargan. Anytime rescheduling. In *Proc. DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control*. Morgan Kaufmann Pub., Nov. 1990.

Session A2: ARTIFICIAL INTELLIGENCE II

Session Chair: Dr. Abe Waksman

N94-34047

Learning procedures from interactive natural language instructions*

Scott B. Huffman and **John E. Laird**
Artificial Intelligence Laboratory
The University of Michigan
1101 Beal Ave.
Ann Arbor, Michigan 48109–2110
huffman@umich.edu

Abstract

Despite its ubiquity in human learning, very little work has been done in artificial intelligence on agents that learn from interactive natural language instructions. In this paper, we examine the problem of learning procedures from interactive, situated instruction, in which the student is attempting to perform tasks within the instructional domain, and asks for instruction when it is needed. We present Instructo-Soar, a system that behaves and learns in response to interactive natural language instructions. Instructo-Soar learns completely new procedures from sequences of instruction, and also learns how to extend its knowledge of previously known procedures to new situations. These learning tasks require both inductive and analytic learning. Instructo-Soar exhibits a multiple execution learning process in which initial learning has a rote, episodic flavor, and later executions allow the initially learned knowledge to be generalized properly.

1 Introduction

The hallmark of universal computation systems is their ability to take instructions. This ability separates computers from other machines, which can perform only a limited number of tasks. Instructability allows computers to perform any of an infinite number of tasks. However, computers take instructions in a way that is radically different from the way humans do. Computers receive instructions in the form of programs. This method of communication from instructor (programmer) to computer is characterized by the following properties.

- **Artificial language.** Programmers must translate knowledge into a language that requires precise artificial terminology and syntax.
- **Unsituated instruction.** The instruction does not occur within the context of the computer attempting to solve a specific problem.
- **Non-interactive instruction.** The instructor determines when and what to instruct without any feedback from the computer.

These properties have a number of implications for the instructor:

1. The instructor must know what procedures are already encoded in the computer, to avoid redundancy and conflicts.
2. The instructor must understand the effects of long sequences of instruction, because a complete instructional sequence must be generated.
3. The instructor must create a procedure that applies in every situation it will be exposed to.
4. The instructor must specify the procedures in complete detail; no steps may be omitted or abstracted.

*This paper also appeared in *Machine Learning: Proceedings of the Tenth International Conference*, ed. Paul E. Utgoff, Amhearst, Mass., June 1993.

In contrast, humans can engage in apprenticeship instruction, in which the student actively tries to acquire knowledge to aid in problem solving. This type of instruction has the following properties:

- **Natural language.** Instructor and student speak the same language, and the language is highly flexible.
- **Situated instruction.** The instructor and student are situated within a specific task. The instructor does not need to predict the effects of long instruction sequences because the student performs the task in response to individual instructions. The instructor needs to produce instructions for only the situation at hand, not for all possible situations. The student can use the situation to disambiguate instruction, and cue the recall of relevant domain knowledge.
- **Interactive instruction.** The student can request help during a task. This frees the instructor from specifying the procedure in full detail; instructions are given when needed. The instructor need not know exactly what the student knows about the task. If the student is unable to fill in missing steps or details, more instruction can be requested.

In this paper, we describe Instructo-Soar, a system that learns procedures from interactive natural language instructions. Instructo-Soar attempts to solve problems within a task domain, and requests instruction when it does not know how to make progress on a problem. The instructions are simple English imperative sentences. They can include commands for primitive or known operators, for complex operators that the system does not know how to apply in the current situation, and for completely new complex operators that the system must learn from scratch. These latter cases lead to a recursive use of instruction where the system learns a hierarchy of operators. Both analytic and empirical learning methods are employed so that after instruction, the system can perform similar tasks (and subtasks) without instruction. Learning of a new procedure is initially by rote, using an episodic memory acquired as a side-effect of natural language comprehension. During later executions of the task, analytic techniques generalize the procedure.

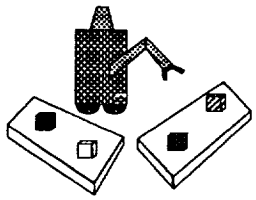
2 Related Work

This work is most closely tied to work on learning from external guidance and advice taking [McCarthy, 1968; Carbonell *et al.*, 1983]. Prior research in these areas has usually emphasized one of the following: natural language instruction, situated instruction, or interactive instruction.

SHRDLU [Winograd, 1972] learned new goal specifications by directly transforming sentences into state descriptions, but did not learn how to perform procedures. Others have learned declarative knowledge bases from natural language (e.g., [Haas and Hendrix, 1983]). A number of recent systems perform in response to natural language input, but do no learning [Vere and Bickmore, 1990; Chapman, 1990; DiEugenio and White, 1992]. Lewis *et al.* [1989] present a system that learns operator sequences from natural language instructions taken in batch form (unsituated and non-interactive). Alterman *et al.* [1991] and Martin and Firby [1991] describe situated systems that recover from execution errors by learning from instruction.

There have been a variety of systems that learn from observation [Redmond, 1989; Segre, 1987; Wilkins, 1988; VanLehn, 1987]. These systems take traces of expert behavior on a specific problem and learn general procedures using analytic techniques such as explanation-based learning [Mitchell *et al.*, 1986]. However, these system do not support interaction with the instructor.

Learning apprentice systems (LAS's) [Mitchell *et al.*, 1990; Kodratoff and Tecuci, 1987; Golding *et al.*, 1987; Laird *et al.*, 1990] extend the work on learning from observation by providing some interactivity: typically, the system suggests steps to the instructor. However, the instructor can only select actions that are directly performable at the current problem state. LAS's cannot take instructions specifying new, unknown actions (that thus must be learned) or actions with unmet preconditions (which the agent may or may not know how to achieve). For example, LEAP's [Mitchell *et al.*, 1990] instructor inputs a complete circuit implementing a desired function, but cannot instruct the system to perform some new, unknown circuit transformation. Since whole circuits are learned for each function, LEAP avoids the problem of an instructor "skipping steps" by specifying operations with unmet preconditions. In addition, learning apprentice systems require that either the termination conditions of the procedure being taught (the goal concept [Mitchell *et al.*, 1986]) are already known, or that the instructor provide a complete description of termination conditions. Finally, these systems typically have no natural language capability.



(a)

Pick up the red block.
  Move to the yellow table.
  Move the arm above the red block.
    Move up.
  Move down.
  Close the hand.
  Move up.
  The operator is finished.

(b)

Figure 1: (a). Initial situation of agent; (b) Instructions to teach operator.

3 Instruction within an autonomous agent

One factor that most previous work on instruction has ignored is the integration of learning from instruction within an autonomous agent. To learn from interactive instruction, an autonomous agent must have general reasoning capabilities, and be able to recognize when its knowledge is insufficient and instruction is needed. Instructions must be assimilated into a possibly large body of existing knowledge, and instructional learning must be smoothly integrated with the agent's other learning and problem solving methods. An agent must maintain its ability to respond to its environment even while accepting instructions, and must be able to apply learning from instruction to a wide variety of tasks.

Supporting these capabilities is dependent in part upon the architecture in which the agent in constructed. We use Soar [Laird *et al.*, 1987] as our underlying architecture. Soar's basic structure provides a framework in which these capabilities can be approached.

In Soar, all activity occurs by applying operators to states within problem spaces, supporting general problem solving and planning. Our instruction learning techniques learn and extend operators, and thus have the potential to be applicable to any problem encoded in Soar. When a Soar agent cannot make progress within a problem space, an impasse arises, and a subgoal is generated to resolve the impasse. Any type of knowledge might be applied within the subgoal, so learning from instruction can co-exist with learning from other knowledge sources, such as experimentation, analogy, etc. Learning occurs through chunking, a form of explanation-based learning, which summarizes the results of subgoal processing, avoiding the impasse in the future. Chunking occurs over all subgoals, so instructional learning can be performed as part of the ongoing activity of the agent, rather than using a separate mechanism that interrupts the normal course of activity.

Instructo-Soar's problem spaces implement an agent with three main categories of knowledge: natural language processing knowledge, originally developed for NL-Soar [Lehman *et al.*, 1991]; knowledge about obtaining and using instruction; and knowledge of the task domain itself. This task knowledge is extended through learning from instruction. Assumed characteristics of the Instructo-Soar agent include:

1. **Relevant relationships.** The agent has knowledge of all of the relevant task relationships, and can derive them from perception.
2. **Primitive operators.** The agent knows a set of primitive operators that it can execute, internally simulate, and map natural language to.
3. **Reading ability.** The agent can read the instructions, even if it has no knowledge of an operation within the current task domain that corresponds to the instruction.
4. **Locality of instruction.** The agent assumes that an instruction applies to the *most recent* unachieved operation.

4 Learning from instruction

Consider the agent and situation shown in Figure 1(a). The agent has primitive operators for moving to tables, opening and closing its hand, and moving its arm up, down, and into relationships with objects (e.g., above blocks). This is the primary domain Instructo-Soar has been applied to; the techniques have also been applied in a more limited way to a flight domain, in which Soar controls a flight simulator and instructions are given for simple maneuvers like taking off [Pearson *et al.*, 1993]. To explain Instructo-Soar's performance, we will use the example of teaching the agent in Figure 1(a) to pick up blocks. Since picking up blocks is not a known operator, when told "Pick up the red block," the agent must learn a new operator.

To learn a new operator, an agent must learn each of the following:

1. **Mapping from natural language:** What instruction(s) map onto the new operator. The mapping allows the agent to select the operator when commanded in the future.
2. **Operator template:** Knowledge of the operator's arguments and how they can be instantiated. For picking up blocks, the agent acquires a new operator with a single argument, which may be instantiated with any block that isn't currently picked up.
3. **Implementation:** How to perform the operator. New operators are built from primitive and/or previously learned operators, so implementation takes the form of a series of sub-operators (e.g., move to the proper table, grasp the block, etc.)
4. **Termination:** Knowledge of when the new operator is achieved. This is the goal concept of the new operator. For "pick up", the termination conditions include holding the desired block, with the arm up.

Instructo-Soar handles simple imperative sentences, and learns a straightforward mapping of an instruction's semantic argument structure to a newly generated operator template. In general, mapping from instructions to task operators and objects can be difficult, as it can require complex natural language comprehension, and possibly reasoning about the task itself [Huffman and Laird, 1992; DiEugenio, 1992].

To learn a general operator implementation from instructions, an agent must determine the proper scope of applicability of each instruction. Some features of the current task and situation are important conditions, while others may be ignored. For example, when told to pick up a red block, does it matter that the block is red? Perhaps, if building a stoplight. But if trying to block open a door, the key feature may be the block's weight. Thus, learning from instruction can involve both generalization (that "red" doesn't matter, although explicitly mentioned) and specialization (that the weight matters, although not mentioned).

To learn general implementations, Instructo-Soar uses explanation-based learning (EBL) as realized by chunking in Soar. Proper generalization requires understanding how each instruction contributes to achieving the goal. However, during the initial execution of the instructions for a new operator, the agent does not know the termination conditions (goal concept) of the operator; therefore, generalization on this basis is impossible. Thus, initial learning is based on a weak inductive step: believing what the instructor says. This learning is rote and overspecific, with an "episodic" flavor. At the end of the initial execution, the termination conditions, or goal concept, of the new operator can be induced. On later executions, the agent *can* form an understanding of how the instructions, recalled from its episodic memory, allow the goal to be reached, using its knowledge of primitive operator effects (domain theory). This allows the implementation sequence to be learned deductively (and generally), based on achievement of the induced goal concept. We will describe the details of the technique using an example.

5 Example

Consider the agent shown in Figure 1(a) being instructed to pick up blocks. The agent is given the instructions shown in Figure 1(b) during the course of performing the task.

5.1 First execution

The agent begins with knowledge of the primitive operators, but no knowledge of the new operator it will be instructed to perform. Following the first execution, the agent must be able to perform at least the exact same task without being re-instructed. Thus, the agent must learn, in some form, *all* of the parts of the new operator, as described in the previous section.

The first instruction given is "Pick up the red block." It is comprehended using Soar's natural language capability, NL-Soar [Lehman *et al.*, 1991], which produces a semantic structure and resolves "the red block" to a block in the agent's environment. However, the semantic structure produced does not correspond to any known operator, indicating that the agent must learn a new operator. Thus, a new, empty operator is generated (e.g., `new-op14`), with an argument structure that directly corresponds to the semantic structure's arguments (here, one argument, `object`). The system learns a mapping from the semantic structure to the new operator, heuristically restricting arguments to be of the same type (e.g., `isa block`) as in the current instruction.

Next, the new operator is selected for execution. Since the agent doesn't know any implementation for the operator, it immediately impasses and asks for further instructions. Each instruction in Figure 1(b)

is given, comprehended and executed in turn. For instance, "Move to the yellow table" is comprehended, mapped to a known operator, and executed. These instructions provide the implementation for the new operator.

The instruction "Move the arm above the red block" provides an example of learning to achieve the preconditions of a known operator. This operation is known, and the agent can perform it when its hand is in the upper plane. However, here the hand is in the lower plane, so the operation cannot be performed directly. Thus, the agent asks for instruction about this operator, and is told to move the arm up. This achieves the precondition, and after moving up the agent has sufficient knowledge to complete the move above operation without further instruction. As a result, a rule is learned that will propose moving up to achieve this precondition in the future. Thus, a known operation has been extended to apply in a new situation; further instruction could extend it even further, for instance allowing the agent to "move above" starting from a state where it's not even next to the table. Note that the interactive nature of instruction means that the instructor *need not know* beforehand whether the agent knows how to apply an operation from the current situation. This recursive instruction of sub-operations could be multiple levels deep, allowing for a great flexibility of instruction sequences, depending what the agent already knows. A simple mathematical analysis shows that for a sequence of only six primitive actions, with preconditions for each action, over 100 possible sequences of interactive instruction are possible [Huffman, 1992]. This contrasts with learning from observation, in which systems learn from observing only the sequence of primitive operations performed to carry out the task.

After completing the "move above" action, the agent continues receiving and executing instructions for the new "pick up" operator. Ultimately, the implementation sequence for "pick up" will be learned at the proper level of generality, based on understanding how each instructed operator leads to successful execution of the new operator. During the initial execution, however, this is impossible, because the goal of this new operator is not yet known. Thus, the agent resorts to *rote learning*, recording exactly what it was told to do, in exactly what situation.

This recording process is not an explicit memorization step; rather, it occurs as a side effect of language comprehension. While reading each sentence, the agent learns a set of rules that encode the sentence's semantic features. The rules help NL-Soar to resolve referents in later sentences (implementing a simple version of Grosz's focus space mechanism [Grosz, 1977]). The rules record each instruction, indexed by the goal it applies to and its place in the instruction sequence. This episodic "case" corresponds to a lock-step, overspecific sequencing of the instructions given to perform the new operator. For instance, the agent encodes that "to pick-up (that is, new-op14) the red block, rb1, I was first told to move to the yellow table, yt1." One issue that arises here is whether and when to generalize the index and information contained within the case. However, at this point any generalization would be purely heuristic, since the agent was unable to explain the various steps of the episode.

Finally, the agent is told "The operator is finished," indicating that the goal of the new operator has been achieved. This triggers the agent to learn termination conditions for the new operator. Learning termination conditions is an inductive concept formation problem. Standard concept learning approaches may be used here; however, typically, an instructor will expect learning within a small number of examples. Currently, the system uses a simple heuristic: it compares the current state to the initial state the agent was in when commanded to perform the new operator. Everything that has *changed* is considered a part of the termination conditions of the new operator. In this case, the changes are that the robot is standing at a table, holding a block, and the block and gripper are both up in the air.

This heuristic forms the system's inductive bias for learning termination conditions. It allows learning from a single example, but is clearly too simple. Conditions that changed may not matter; e.g., perhaps it doesn't matter to picking up blocks that the robot ends up at a table. Unchanged conditions may matter; e.g., if learning to build a "stoplight", block colors are important.

Instructo-Soar performs this induction by EBL over an overgeneral theory (as, e.g., [Miller and Laird, 1991; VanLehn *et al.*, 1990; Rosenbloom and Aasman, 1990]). Although not sophisticated here, this type of inductive learning has the advantage that the agent can alter the bias to reflect other available knowledge. This might include more instruction (e.g., simply asking which features are relevant [Laird *et al.*, 1990]); analogy to other known operators (e.g., pick up actions in related domains), domain heuristics, etc.

On the first pass, then, the agent:

- Carries out a sequence of instructions achieving a new operator.
- Learns a new operator template.

- Learns the mapping from natural language to the new operator.
- Learns a rote execution sequence for the new operator.
- Induces the termination conditions of the new operator.

Since the agent has learned all of the necessary parts of an operator, it will be able to perform the same task again without instruction. However, since the implementation of the operator is rote, it can only perform the *exact* same task. It has not learned generally how to pick up blocks yet.

Since the goal is now known, the system could explain and generalize the instruction sequence directly after the first execution. This is a reasonable possibility, but the multi-step simulation required has two disadvantages:

1. The agent's ongoing performance of the tasks at hand (either by acting or by taking more instruction) is temporarily suspended. This could be awkward if instruction of the new procedure being simulated is nested within the instructions for larger tasks that must still be completed, or if these tasks have temporal constraints.

2. The multiple step simulation is susceptible to compounding of domain theory errors. That is, a significant error in simulating any step of the procedure (or the interaction of multiple small errors) can lead to an incomplete or incorrect explanation of goal achievement. Simulating to explain each *individual* instruction, as described below, avoids this problem because each successive simulation begins from the current external state, which reflects the true effects of the previous instructions.

Thus, we have opted for generalizing on future executions.

5.2 Later executions

Later, in the same situation the agent is again asked to pick up the red block. The agent selects the newly learned operator, and then reaches an impasse because it does not yet know the general implementation sequence for the operator (how to pick up blocks in the general case). Here, the agent attempts to recall instructions it was given during the first execution. It retrieves, instruction by instruction, the rote case it learned previously.

After each retrieval, before carrying out the instruction in the external world, the agent attempts to explain to itself *why* the instructed operator leads to achievement of the higher-level goal of picking up the block. This explanation attempt takes the form of an internal simulation. Starting from the current world state, the agent internally simulates the recalled operator. Thus, the *situatedness* of the instruction plays a key role in the learning process, because the current situation grounds the explanation. The agent continues to simulate operators until it either reaches its higher-level goal (internally) or does not know what to do next. If the goal is reached, the path taken to the goal comprises an explanation of how the recalled operator leads to goal achievement.

From this explanation, the system learns a general rule that proposes the recalled operator under the right conditions. The rule both generalizes and specializes the original instruction. In "move to the yellow table"'s case, the color of the table is generalized away, because it was not critical for achievement of the goal, while the fact that the table has the block to be picked up on it is included in the proposal rule's conditions.

After learning the complete general implementation, the agent will perform the task without reference to the rote case. If asked to "Pick up the green block," `new-op14` is selected and instantiated with the green block as its argument. Then, the general sub-operator proposal rules for `new-op14` fire one by one, implementing the operator, until finally the termination conditions are recognized and the operator is terminated. Since the general proposal rules test state conditions directly, the agent can perform the task starting from any state along the implementation path, and can react to unexpected conditions (e.g., another robot stealing the block). In contrast, the rote implementation had to be performed from the same initial state each time, and its steps were not conditional on the state.

6 Results

In the robotic domain described earlier, Instructo-Soar has been applied to learn a hierarchy of task operators, shown in Figure 2. The system read 24 natural language instructions (14 unique sentences) and learned 1357

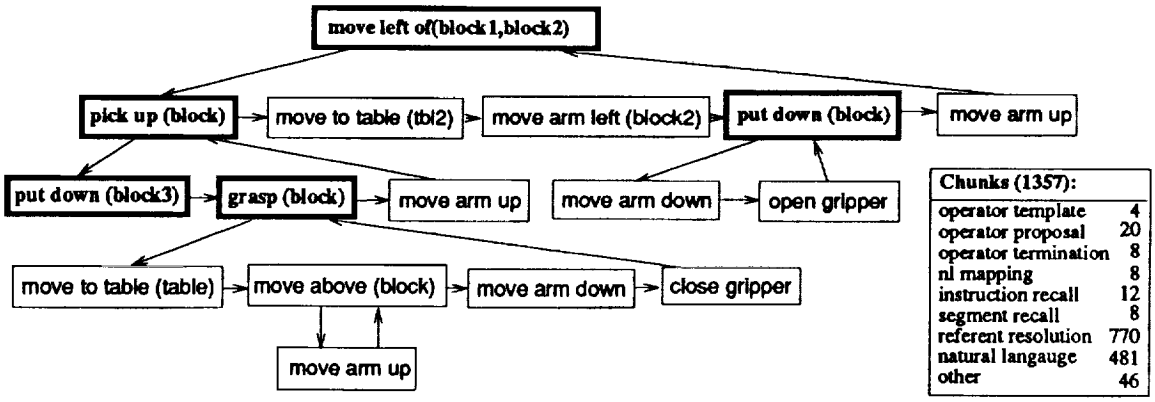


Figure 2: A hierarchy of operators learned by Instructo-Soar. Primitive operators are shown in light print; learned operators are in boldface.

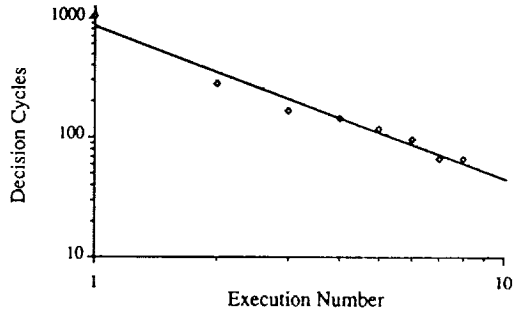


Figure 3: Decision cycles vs. learning trial for executing "pick up".

chunks broken down as shown in the figure. It learned four completely new operators (shown in bold), how to achieve preconditions for a known operator (`move above`), and the extension of a new operator (extending "pick up" to work if the robot already is holding a block after initially learning "pick up" when the gripper was empty). This hierarchy can be taught using many different instruction orderings. For instance, new operators that appear as sub-operators (e.g., `grasp`) can be taught either before teaching higher operators (e.g., `pick up`), or during teaching of them. If during, the agent recursively requests instruction for the lower new operator. Thus the instructor need not know whether the agent knows a procedure before commanding it. This is an important advantage of interactive instruction for autonomous agents, which may have large amounts of knowledge. Similarly, the instructor may suggest an action that has unmet preconditions (thus skipping steps in the instruction sequence), assuming the agent knows how to achieve them before performing the action. If the agent does not, it can request more instruction, and learn how to achieve the preconditions.

Instructo-Soar exhibits a number of interesting learning characteristics:

- **Multiple recall strategies.** Instructo-Soar has two strategies it can use in recalling past instructions. After recalling and internally simulating an instructed operator, the agent still may not know how that operator leads to the goal. At this point, the agent may terminate its internal simulation, and carry out the recalled operator in the external world. This is a *single recall* strategy, which is appropriate when the agent is under pressure to act quickly. Alternatively, the agent may attempt to recall further instructions, simulating each in turn, until the higher-level goal is reached. This is a *multiple recall* strategy, which leads to faster learning, but is more susceptible to errors in the agent's domain theory (primitive operator knowledge), as described above.
- **Bottom-up learning** (single recall strategy). Limiting recall to a single step allows only a single sub-operator per execution to be generalized. Generalized learning begins at the end of the implementation sequence and moves towards the beginning. As Figure 3 shows for learning "pick up", the resulting learning curve closely approximates the power law of practice [Rosenbloom and Newell, 1986] ($r = 0.98$).

C-5.

- **Effectiveness of hierarchical instruction** (single recall strategy). Due to the bottom-up effect, the system learns more quickly when taught hierarchical organizations than flat sequences. General learning for an N step operator takes N executions using a flat instruction sequence. Taught hierarchically as $\sqrt{N}$ sub-operators with $\sqrt{N}$ steps each, only $\sqrt{N}$ executions are required for full general learning. Hierarchical organization has the additional advantage that more operators are learned that can be used in future instruction.
- **Degradation without domain theory**. If the agent does not have knowledge of the primitive operators' effects, learning degrades to rote learning. This appears to be consistent with psychological research showing that subjects given procedural instructions learn and perform better when they have a domain model [Kieras and Bovair, 1984].

7 Conclusion

We have described Instructo-Soar, an agent that learns and extends procedures by receiving interactive, situated natural language instructions. The agent learns completely new operators: preconditions, implementation, and termination conditions (goal concept), in contrast to learning apprentice systems, which learn only implementations, and preconditions of those implementations, for known operators. New operators learned by Instructo-Soar may be specified in later instructions for other operators, leading to learning of operator hierarchies. From the initial execution of a new operator, the agent learns a rote, overspecific execution sequence, and induces termination conditions. On later executions, the execution sequence is generalized by using internal simulation to explain each instruction.

Instructo-Soar can be extended in a number of directions. In addition to positive imperative sentences, we are currently investigating learning from other types of instructions, such as positive and negative constraints, conditionals, and actions with monitoring conditions [Huffman and Laird, 1992]. The difference-of-states method used to induce operator termination conditions is being extended to allow instructor feedback about the induced conditions. Finally, allowing weaker forms of explanation, such as analogy and heuristic causality theories (e.g., [Pazzani, 1991; VanLehn *et al.*, 1992; Schank and Leake, 1989]), would lead to more graded degradation of learning with domain theory incompleteness. These types of explanation might also lead the agent to alter its basic domain theory, for instance by inferring previously unknown affects of primitive actions.

Acknowledgments

This research was sponsored by NASA/ONR under contract NCC 2-517. The authors wish to thank Paul Rosenbloom, Jill Lehman, Rick Lewis, and Randy Jones for valuable comments on earlier drafts.

References

[Alterman *et al.*, 1991] Richard Alterman, Roland Zito-Wolf, and Tamitha Carpenter. Interaction, comprehension, and instruction usage. Technical Report CS-91-161, Computer Science Department, Brandeis University, July 1991.

[Carbonell *et al.*, 1983] Jaime G. Carbonell, R. S. Michalski, and T. M. Mitchell. An overview of machine learning. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An artificial intelligence approach*. Morgan Kaufmann, 1983.

[Chapman, 1990] David Chapman. *Vision, Instruction, and Action*. PhD thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, April 1990.

[DiEugenio and White, 1992] B. DiEugenio and M. White. On the interpretation of natural language instructions. In *Proceedings COLING 92*, July 1992.

[DiEugenio, 1992] B. DiEugenio. Understanding natural language instructions: The case of purpose clauses. In *Proceedings of Annual Meeting of the ACL*, July 1992.

[Golding *et al.*, 1987] A. Golding, P. S. Rosenbloom, and J. E. Laird. Learning search control from outside guidance. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 334–337, August 1987.

[Grosz, 1977] Barabara J. Grosz. *The Representation and use of focus in dialogue understanding.* PhD thesis, University of California, Berkeley, 1977.

[Haas and Hendrix, 1983] Norman Haas and Gary G. Hendrix. Learning by being told: Acquiring knowledge for information management. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An artificial intelligence approach.* Morgan Kaufmann, 1983.

[Huffman and Laird, 1992] Scott B. Huffman and John E. Laird. Dimensions of complexity in learning from interactive instruction. In Jon Erikson, editor, *Proceedings of Cooperative Intelligent Robotics in Space III, SPIE Volume 1829*, November 1992.

[Huffman, 1992] Scott B. Huffman. An analysis of instruction sequences. Artificial Intelligence Laboratory, University of Michigan, 1992.

[Kieras and Bovair, 1984] David E. Kieras and Susan Bovair. The role of a mental model in learning to operate a device. *Cognitive Science*, 8:255–273, 1984.

[Kodratoff and Tecuci, 1987] Yves Kodratoff and Gheorghe Tecuci. DISCIPLE-1: Interactive apprentice system in weak theory fields. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 271–273, August 1987.

[Laird *et al.*, 1987] John E. Laird, Allen Newell, and Paul S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[Laird *et al.*, 1990] John E. Laird, Michael Hucka, Eric S. Yager, and Christopher M. Tuck. Correcting and extending domain knowledge using outside guidance. In *Proceedings of the Seventh International Conference on Machine Learning*, 1990.

[Lehman *et al.*, 1991] Jill Fain Lehman, Richard L. Lewis, and Allen Newell. Natural language comprehension in Soar: Spring 1991. Technical Report CMU-CS-91-117, School of Computer Science, Carnegie Mellon University, March 1991.

[Lewis *et al.*, 1989] Richard L. Lewis, Allen Newell, and Thad A. Polk. Toward a Soar theory of taking instructions for immediate reasoning tasks. In *Proceedings of the Annual Conference of the Cognitive Science Society*, August 1989.

[Martin and Firby, 1991] Charles E. Martin and R. James Firby. Generating natural language expectations from a reactive execution system. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 811–815, August 1991.

[McCarthy, 1968] J. McCarthy. The advice taker. In M. Minsky, editor, *Semantic Information Processing*, pages 403–410. MIT Press, 1968.

[Miller and Laird, 1991] C. Miller and J. E. Laird. A constraint-motivated lexical acquisition model. In *Proceedings of the 13th Annual Conference on the Cognitive Science Society*, pages 827–831, Boston, 1991.

[Mitchell *et al.*, 1986] Tom M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1, 1986.

[Mitchell *et al.*, 1990] T. M. Mitchell, Sridhar Mahadevan, and Louis I. Steinberg. LEAP: A learning apprentice system for VLSI design. In Yves Kodratoff and R. S. Michalski, editors, *Machine Learning: An artificial intelligence approach, Vol. III.* Morgan Kaufmann, 1990.

[Pazzani, 1991] Michael Pazzani. Learning to predict and explain: An integration of similarity-based, theory driven, and explanation-based learning. *Journal of the Learning Sciences*, 1(2):153–199, 1991.

[Pearson *et al.*, 1993] Douglas J. Pearson, Scott B. Huffman, Mark B. Willis, John E. Laird, and Randolph M. Jones. Intelligent multi-level control in a highly reactive domain. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, Pittsburgh, PA., February 1993.

[Redmond, 1989] Michael Redmond. Combining case-based reasoning, explanation-based learning and learning from instruction. In *Proceedings of the International Workshop on Machine Learning*, pages 20–22, 1989.

[Rosenbloom and Aasman, 1990] Paul S. Rosenbloom and Jans Aasman. Knowledge level and inductive uses of chunking (ebl). In *Proceedings of the National Conference on Artificial Intelligence*, 1990.

[Rosenbloom and Newell, 1986] Paul S. Rosenbloom and Allen Newell. The chunking of goal hierarchies: A generalized model of practice. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An artificial intelligence approach, Volume II.* Morgan Kaufmann, 1986.

[Schank and Leake, 1989] Roger C. Schank and David B. Leake. Creativity and learning in a case-based explainer. *Artificial Intelligence*, 40:353–385, 1989.

[Segre, 1987] Alberto Maria Segre. A learning apprentice system for mechanical assembly. In *Third IEEE Conference on Artificial Intelligence for Applications*, pages 112–117, 1987.

[VanLehn *et al.*, 1990] K. VanLehn, W. Ball, and B. Kowalski. Explanation-based learning of correctness: Towards a model of the self-explanation effect. In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, pages 717–724, 1990.

[VanLehn *et al.*, 1992] Kurt VanLehn, Randolph M. Jones, and Michelene T. H. Chi. A model of the self-explanation effect. *Journal of the learning sciences*, 2(1):1–59, 1992.

[VanLehn, 1987] Kurt VanLehn. Learning one subprocedure per lesson. *Artificial Intelligence*, 31(1):1–40, 1987.

[Vere and Bickmore, 1990] Steven Vere and Timothy Bickmore. A basic agent. *Computational Intelligence*, 6:41–60, 1990.

[Wilkins, 1988] David C. Wilkins. Knowledge base refinement using apprenticeship learning techniques. In *Proceedings of the National Conference on Artificial Intelligence*, pages 646–651, 1988.

[Winograd, 1972] Terry Winograd. *Understanding Natural Language.* Academic Press, New York, 1972.

N94-34048

Fuzzy Logic, Neural Networks, and Soft Computing

Lofti A. Zadeh
University of California
Berkeley, CA

The past few years have witnessed a rapid growth of interest in a cluster of modes of modeling and computation which may be described collectively as soft computing. The distinguishing characteristic of soft computing is that its primary aims are to achieve tractability, robustness, low cost, and high MIQ (machine intelligence quotient) through an exploitation of the tolerance for imprecision and uncertainty. Thus, in soft computing what is usually sought is an approximate solution to a precisely formulated problem or, more typically, an approximate solution to an imprecisely formulated problem. A simple case in point is the problem of parking a car. Generally, humans can park a car rather easily because the final position of the car is not specified exactly. If it were specified to within, say, a few millimeters and a fraction of a degree, it would take hours or days of maneuvering and precise measurements of distance and angular position to solve the problem. What this simple example points to is the fact that, in general, high precision carries a high cost. The challenge, then, is to exploit the tolerance for imprecision by devising methods of computation which lead to an acceptable solution at low cost. By its nature, soft computing is much closer to human reasoning than the traditional modes of computation. At this juncture, the major components of soft computing are fuzzy logic (FL), neural network theory (NN), and probabilistic reasoning techniques (PR), including genetic algorithms, chaos theory, and parts of learning theory. Increasingly, these techniques are used in combination to achieve significant improvement in performance and adaptability. Among the important application areas for soft computing are control systems, expert systems, data compression techniques, image processing, and decision support systems. It may be argued that it is soft computing – rather than the traditional hard computing – that should be viewed as the foundation for artificial intelligence. In the years ahead, this may well become a widely held position.

N94-34049

Hybrid Knowledge Systems *

V.S. Subrahmanian†

Abstract

In this paper, we describe an architecture, due to the author, called *hybrid knowledge systems* (HKS, for short) that can be used to inter-operate between (1) a specification of the control laws describing a physical system (in particular, this could include specifications such as those of Brockett and/or Nerode and Kohn, but is not limited to those), (2) a collection of databases, knowledge bases and/or other data structures reflecting information about the world in which the physical system being controlled resides, (3) observations (e.g. sensor information) from the external world, and (4) actions that must be taken in response to external observations.

1 Introduction

Deductive databases that interact with, and are accessed by, reasoning agents in the real world (such as logic controllers in automated manufacturing, weapons guidance systems, aircraft landing systems, land-vehicle maneuvering systems, and air-traffic control systems) must satisfy a number of diverse, and often conflicting criteria. In this paper, we will describe a software architecture called *hybrid knowledge systems* (HKS, for short) that supports intelligent real-time reasoning in domains such as control systems. In particular, we will show how, given the physical equations governing the dynamics of a control system, as well as the control laws governing the application of control actions, it is possible for our framework to be used as a platform for developing an intelligent, real-time control system. In other words, this platform enables a smooth integration of the knowledge of a control engineer, and the database technology in the HKS system. In particular, this platform enables HKSs to act as a mediator between database systems, and methods for specifying the dynamics of hybrid control systems (e.g. the frameworks of Brockett [2] and/or the Kohn-Nerode framework).

2 The HKS Architecture for Intelligent Control

In [10], Subrahmanian has outlined an architecture for supporting intelligent real-time reasoning systems. This architecture, known as *hybrid knowledge systems* (HKS, for short) is

*This research was supported by the Air Force Office of Scientific Research under Grant Nr. F49620-93-1-0065 and by the Army Research Office under grant DAAL-03-92-G-0225.

†Institute for Advanced Computer Studies, Institute for Systems Research and Department of Computer Science, University of Maryland, College Park, Maryland 20742. E-mail: vs@cs.umd.edu

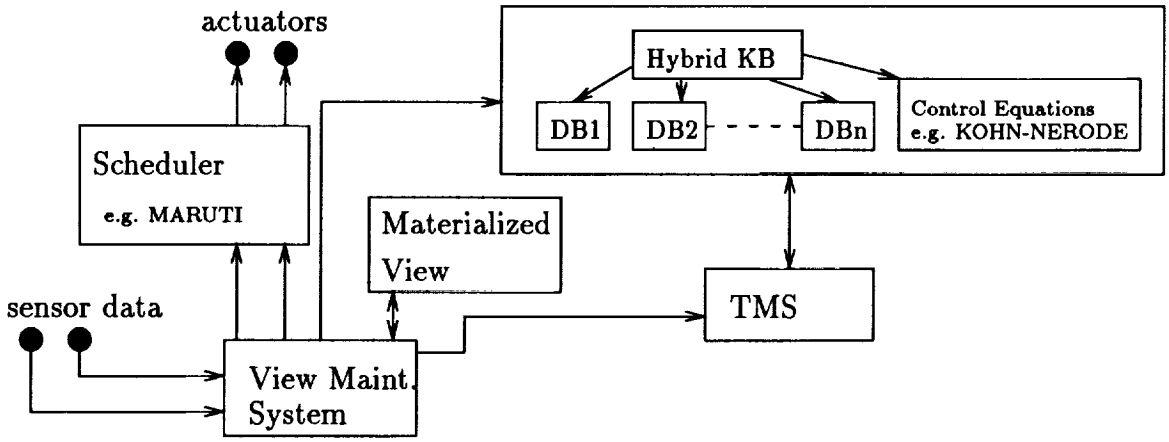


Figure 1: Architecture for Intelligent Support for Real-Time Control Systems

built upon integrating multiple data sources (e.g. sensors), knowledge sources (e.g. knowledge bases, and databases), data structures, and constraint systems. As differential equations are just constraints, the control axioms, and the computations involved in control systems can be represented in the HKS architecture. We describe below, the individual components of the HKS architecture, using the Cruise missile example introduced earlier to illustrate the basic ideas.

2.1 A Snapshot at Time 0

At time 0, the hybrid knowledge base (to be described in detail in Section 2.3) integrates a number of databases, and data domains – one of these consists of a set of numerical differential equations reflecting the dynamics of the physical system being controlled. For example, if we wish to control a missile, this set of equations reflects the control laws used to guide the missile. For those readers who are not control engineers, the dynamics of a control system (e.g. a missile) are specified by (multiple) sets of differential equations reflecting different trajectories (e.g. motions of the missile) of the system being controlled. Each of these sets of equations is called a control law. If one wants to change the trajectory of the system (e.g. the directionality of the missile), one must vary the control law currently being used and determine/specify the time for which that control law is applied.

The initial trajectory of the control system is known, and is denoted by Traj(0) – intuitively, this is an assignment of values to all variables that are used to describe the parameters of the system.

2.2 A Snapshot at Time $t > 0$

Suppose t is an instant of time that occurs during the working of the physical system being controlled. In a missile control example, for instance, t may be a point of time after the

missile has been launched, but before it has completed its mission.

At time t, the HKS architecture would:

- Maintain a small set of facts – this set of facts reflects the current "state" of the environment. In the missile control example, this set of facts includes the position, $\mathsf{POS}(t)$, of the missile, and the position, $(x'(t), y'(t), z'(t))$ of the target at time t. This set of facts is called a *materialized view*. It may, or may not, be consistent with the (relatively static) set of databases integrated by the hybrid knowledge base.
- In addition, at time t, new information may come in, specifying that the the actual values of the variables involved has changed from $\mathsf{Traj}(t)$ to $\mathsf{Traj}(t+1)$. For example, when considering missile control, this may reflect the fact that target has moved from its previous location to a new location.
- Using this information (which reflects a *request to update* the materialized view), the rules in the hybrid knowledge base are used to incrementally determine which set, Act of actions (selected from an available set of actions that the control system can execute) should be performed. Using a specification of the control laws (that will, presumably, be provided by control engineers), the HKS will use these actions to determine the new trajectory. Note that the control laws reside in one of the domains integrated by the hybrid knowledge base[1] and we are not generating them on the fly in real-time; rather, we are selecting certain (possibly parametrized) control laws to apply using the rules in the HKS.

In the rest of this section, we will: (1) explain the basic ideas behind the hybrid knowledge base paradigm [10, 8], and (2) show how control systems can be modeled using the architecture given here, and (3) explain what a materialized view is.

2.3 The Hybrid Knowledge Base Component

Nerode and Subrahmanian have introduced the concept of a hybrid knowledge base for integrating information in multiple data structures and multiple database paradigms. Key to the definition of hybrid knowledge bases is that of a *constraint domain*, described below.

2.3.1 Constraint Domains

Definition 1 Suppose S is a set. The *function space* generated by S, denoted $\mathbf{Func}(S)$, is the smallest set satisfying the following conditions:

1. $S \in \mathbf{Func}(S)$
2. for all integers $n, m \geq 1$, every function $f : S^n \longrightarrow S^m$ is in $\mathbf{Func}(S)$ and
3. if $\emptyset \neq G_1, G_2 \subseteq \mathbf{Func}(S)$, then every function $g : G_1 \rightarrow G_2$ is in $\mathbf{Func}(S)$.

[1] The role of the TMS, or Truth Maintenance System has not been elucidated here. It will be discussed later, in Section 2.5.

Basically, **Func**(S) contains not only all functions from S^i to S^j for all $i, j \geq 1$, but also all *functions* on *sets* of functions. For example, if S is the set $\mathbf{R}$ representing the reals, then the definite integral

$$\int_a^b f(x)dx$$

may be viewed as a higher order function INT that takes as input, a function f, and the real numbers a and b, and returns as output, a real number, i.e.

$$INT(f, a, b) = f^*(b) - f^*(a)$$

where f^* is the integral of f. In the special case when $f(x) = (3x^2 + 4x + 5)$,

$$INT(f, a, b) = (b^3 + 2b^2 + 5b) - (a^3 + 2a^2 + 5a).$$

Definition 2 A *constraint domain* Σ is a triple (D, F, R) where:

- D is a non-empty set called the "domain of discourse" and
- $F \subseteq$ **Func**(D) and
- R is a set of binary relations on $D \cup$ **Func**(D).

Intuitively, a constraint domain Σ specifies a set D representing the domain of discourse over which we are working. The set F is the set of functionals (over the domain D) that are of interest, and the set R of relations over the functionals represents the kinds of relations we are interested in.

For example, if $\mathbf{R}$ (the set of reals) is our domain of discourse, we may have a relation $r \in R$ that says that if $f, g : \mathbf{R} \rightarrow \mathbf{R}$, then

$$f\, r\, g \leftrightarrow (\forall x \in \mathbf{R}) f(x) \leq g(x).$$

In this case, for the pair (f, g) to be in the relation r, both f and g must be unary functions on $\mathbf{R}$ and they must satisfy the above condition of "belowness."

As another example, we may consider equality as our relation, and express differential equations such as:

$$3\frac{dy}{dx} + 4 = x.$$

Here, just as integrals were considered to be functionals, differential operators may also be regarded as functionals.

The reader will notice that according to this definition, a constraint domain is a very general structure. This is indeed the case, and it was proved in [8] that many useful structures such as quadtrees, R-trees, relational databases, object oriented databases, etc. can be viewed as constraint domains.

Given a constraint model $\Sigma = (D, F, R)$, we associate with each element $d \in D$, a symbol d_s. With each $f \in F$, we associate a symbol f_s, and with each relation $r \in R$, a symbol r_s.

Definition 3 Given a constraint model $\Sigma = (D, F, R)$, we may define an atomic constraint as follows: if $r \in R$ is a relation, and α, β are in $\mathbf{Func}(S)$, then $(\alpha_s\, r_s\, \beta_s)$ is an *atomic* constraint. A *constraint* is defined as follows:

- Every atomic constraint is a constraint.
- If C is a constraint, then $\neg C$ is a constraint.
- If C, D are constraints, then $(C \,\&\, D)$ and $(C \vee D)$ are constraints.
- Nothing else is a constraint.

Definition 4 Given a constraint model $\Sigma = (D, F, R)$, and a constraint C, we may define the *satisfaction* of C by Σ, denoted $\Sigma \rhd C$, as follows:

- If C is the atomic constraint $(\alpha_s\, r_s\, \beta_s)$, then $\Sigma \rhd C$ iff $(\alpha\, r\, \beta)$, i.e. the pair $(\alpha, \beta) \in r$.
- If C is the constraint $\neg D$, then $\Sigma \rhd C$ iff it is not the case that $\Sigma \rhd D$.
- If C is the constraint $(D \,\&\, E)$, then $\Sigma \rhd C$ iff $\Sigma \rhd D$ and $\Sigma \rhd E$.
- C is the constraint $(D \vee E)$, then $\Sigma \rhd C$ iff $\Sigma \rhd D$ or $\Sigma \rhd E$.

Thus, for every constraint C, and any constraint model $\Sigma = (D, F, R)$, $\Sigma \rhd C$ or $\Sigma \rhd \neg C$.

2.3.2 Hybrid Knowledge Bases

An *annotation* is a pair $[u, t]$ where u is a term ranging over the unit interval (i.e. either a real number in the unit interval, a variable ranging over the unit interval, or a complex term consisting of a unit-interval valued function applied to sub-terms that range over the unit interval) and t is a term ranging over *sets* of non-negative real numbers (i.e. t is either a set of non-negative real numbers, or t is a variable ranging over sets of non-negative real numbers, or t is a complex term consisting of a non-negative real-valued set-valued function applied to sub-terms of the same type). A natural ordering $\preceq$ on variable-free annotations is the pointwise ordering induced by $\leq$ and $\subseteq$. In other words,

$$[u, t] \preceq [u', t'] \text{ iff } u < u' \text{ and } t \subset t'.$$

Definition 5 If A is a usual atomic formula of predicate calculus (built out of ordinary variables, predicate symbols, and constant symbols) and $[u, t]$ is an annotation, then $A : [u, t]$ is an *annotated atom.* An annotated atom containing no occurrences of object variables is *ground.*

Intuitively, the annotated atom $A : [u, t]$ says that "A is true with certainty at least u at all time points in the set t." When $u = 1$ and $t = \mathrm{R}^+$, then we will simply write A instead of writing $A : [u, t]$.

Definition 6 A *constrained-clause* is a sentence of the form

$$A : [u_0, t_0] \leftarrow \Xi_1, \ldots, \Xi_m \parallel B_1 : [u_1, t_1] \,\&\, \ldots \,\&\, B_n : [u_n, t_n]$$

where $A, B_1, \ldots, B_n$ are atoms of the language L, Ξ_i is a constraint over Σ_i, and

$$A : [u_0, t_0] \leftarrow B_1 : [u_1, t_1] \,\&\, \ldots \,\&\, B_n : [u_n, t_n]$$

is an annotated clause. Ξ is called the *constraint part* of the above clause, and $A : [u_0, t_0] \leftarrow B_1 : [u_1, t_1] \,\&\, \ldots \,\&\, B_n : [u_n, t_n]$ is called the *annotated clause part* of the above formula.

2.3.3 Using Hybrid KBs to Generate the Snapshot at Time $t > 0$

The main purpose for which the hybrid knowledge base will be used is to determine, based upon changes in the trajectory of the system, what the new orientation of the missile ought to be – in particular, the hybrid KB must specify which of the available control actions should be applied.

We assume that there is a predicate, called change that specifies the change in the trajectory. Thus, at any given point in time, a fact of the form change(−) is added as an update to the materialized view, and a set of actions must be generated by the hybrid knowledge base. Observe that at time t, we must compute:

- What controls to apply, and
- How long these controls must be applied for.

2.4 The View Maintenance Component

For the purposes of this paper, a *view* is just a hybrid knowledge base. *Materialization* of a view (i.e. of a hybrid knowledge base) refers to the task of computing, and storing, parts of the unique least Herbrand model of the hybrid KB. As all hybrid KBs are just sets of constrained clauses, which are negation-free ([8] also studies the case when nonmonotonic modes of negation are present), such a unique least model is guaranteed to exist by results in [8]. Index structures can be built on the materialized view. Consequently, database accesses to materialized view tuples is much faster than by recomputing the view. Materialized views are especially useful for providing intelligent support to real-time control systems for the following reasons:

1. at time t, determining the current trajectory is a constant time retrieval operation,
2. the new trajectory at time $t+1$ can be viewed as an *update* to the view, saying that the atom Traj($\langle new \rangle$) should be inserted into the view, and the atom Traj($\langle old \rangle$) should be deleted from the view.
3. Subsequently, using incremental view maintenance techniques such as those described by Gupta, Mumick and Subrahmanian [3], these updates can be easily incorporated into the materialized view.

2.5 The Truth Maintenance Component

The primary reason for using view maintenance algorithms in real-time is that they have much better computational properties than truth maintenance algorithms. The view maintenance algorithms in [3] all have linear-time data complexity ; however, even for definite logic programs, truth maintenance is known to be NP-hard. Even though specific types of instances of NP-hard problems can, and often are, solved efficiently, it turns out (cf. [3]) that view maintenance is always computationally easier than truth maintenance. The reason for this is that if $T \models A$ (i.e. a set T of formulas has A as a logical consequence), and we wish to update T by asserting $\neg A$, then truth maintenance systems attempt to do two things: (1) prevent the derivability of A from T, and (2) attempt to establish which formulas that were provable from T are no longer provable from T (based on A being "false" as a result of the update). View maintenance algorithms only perform (2), and do not account for (1).

Our architecture separates truth maintenance and view maintenance into two phases. When real-time performance is desired and time is at a premium, view maintenance is performed; when additional time is available to analyze the cause of discrepancies between sensor information and the materialized view, then the hybrid knowledge base can be changed so as to ensure consistency with the materialized view; that is, truth maintenance is performed off-line (or when slack-time is available on the processors), view maintenance is performed in real-time. Hybrid Knowledge Systems present an architecture that supports intelligent real-time reasoning. In short, the HKS architecture shows how view maintenance techniques such as those of Gupta, Mumick and Subrahmanian [3], view materialization techniques such as those of Bell, Nerode, Ng and Subrahmanian [1], truth maintenance techniques, and efficient database mediation techniques [9, 8], and specification of control laws such as those of Brockett [2] and Kohn and Nerode [7].

3 Conclusions

In this paper, we have described the notion of a hybrid knowledge system, and shown how the HKS architecture can be used to support and seamlessly integrate the modes of computation required to provide intelligent support to real-time systems such as control systems. Complex reasoning systems of this kind need to be able to reason with multiple representations of data, knowledge, and reasoning paradigms. They must also have a facility whereby different models of control (e.g. [2, 7]) may be incorporated. The HKS paradigm provides the expressive power and facilities required for this purpose through the mechanism of hybrid knowledge bases [8, 10]. In addition to performing such modes of reasoning, real-time performance is also required of such systems in the presence of dynamic changes to the external world. We have shown how view maintenance algorithms in databases can be used to elegantly capture these phenomena.

We are currently developing two applications of HKSs to real-time control systems – one is in mobile robotics [4] at NIST, and the other is in missile control.

References

[1] C. Bell, A. Nerode, R. Ng and V.S. Subrahmanian. (1991) *Mixed Integer Programming Methods for Computing Nonmonotonic Deductive Databases*, to appear in: Journal of the ACM.

[2] R. Brockett. (1993) *Hybrid Models for Motion Control Systems*, to appear in: Perspectives in Control (eds. H.L. Trentelman and J.C. Willems), Birkhauser.

[3] A. Gupta, I. S. Mumick and V.S. Subrahmanian. (1993) *Maintaining Views Incrementally*, Proc. 1993 ACM SIGMOD Intl. Conf. on Management of Data, pps 157–166.

[4] J. Horst, E. W. Kent, H. Rifky and V.S. Subrahmanian. (1993) *Intelligent, Real-Time Robotic Reasoning with Hybrid Knowledge Systems*, draft manuscript.

[5] M. Kifer and V.S. Subrahmanian. (1989) *Theory of Generalized Annotated Logic Programming and its Applications*, Proc. 1989 North American Conf. on Logic Programming, MIT Press.

[6] M. Kifer and V. S. Subrahmanian. (1992) *Theory of Generalized Annotated Logic Programming and its Applications*, Journal of Logic Programming, Vol. 12, 4, pps 335–368, 1992.

[7] W. Kohn and A. Nerode. (1992) *An Autonomous Systems Control Theory: An Overview*, Invited Address. Proc. 1992 IEEE Symp. on Computer-Aided Control Systems Design, pps 204–210.

[8] J. Lu, A. Nerode, J. Remmel and V.S. Subrahmanian. (1993) *Towards a Theory of Hybrid Knowledge Bases*, Univ. of Maryland CS-TR-3037. Submitted for publication.

[9] V.S. Subrahmanian. (1992) *Amalgamating Knowledge Bases*, Univ. of Maryland CS-TR-2949, August 1992. Submitted to ACM Trans. on Database Systems, Aug. 1992. Revised May 1993.

[10] V.S. Subrahmanian. (1993) *Hybrid Knowledge Bases for Intelligent Reasoning Systems*, Invited Address, Proc. 8th Italian Conf. on Logic Programming (ed. D. Sacca), pps 3–17, Gizzeria, Italy, June 1993.

TWO FRAMEWORKS FOR INTEGRATING KNOWLEDGE IN INDUCTION

Paul S. Rosenbloom
Information Sciences Institute &
Department of Computer Science
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292

Haym Hirsh
Department of Computer Science
Hill Center, Busch Campus
Rutgers University
New Brunswick, NJ 08903

William W. Cohen
AT&T Bell Laboratories
600 Mountain Avenue
Room 3C-412A
Murray Hill, NJ 07974

Benjamin D. Smith
Information Sciences Institute &
Department of Computer Science
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292

ABSTRACT

The use of knowledge in inductive learning is critical for improving the quality of the concept definitions generated, reducing the number of examples required in order to learn effective concept definitions, and reducing the computation needed to find good concept definitions. Relevant knowledge may come in many forms (such as examples, descriptions, advice, and constraints) and from many sources (such as books, teachers, databases, and scientific instruments). How to extract the relevant knowledge from this plethora of possibilities, and then to integrate it together so as to appropriately affect the induction process is perhaps the key issue at this point in inductive learning. Here we focus on the integration part of this problem; that is, how induction algorithms can, and do, utilize a range of extracted knowledge. Preliminary work on a transformational framework for defining knowledge-intensive inductive algorithms out of relatively knowledge-free algorithms is described, as is a more tentative problem-space framework that attempts to cover all induction algorithms within a single general approach. These frameworks help to organize what is known about current knowledge-intensive induction algorithms, and to point towards new algorithms.

INTRODUCTION

Inductive learning is a process whereby a definition of a concept is derived from a set of positive, and sometimes negative, examples of the concept. Key issues in inductive learning are the accuracy of the resulting definition – that is, the error rate it yields in classifying new examples – and the resources required to generate the definition (in terms of the number of examples and/or the amount of time and space needed).

The single most promising route towards reducing both the error rate and resource usage of inductive learning is to utilize whatever additional knowledge is available beyond the examples; that is, to convert induction from a weak to a strong method. However, to do this, the relevant knowledge must first be extracted from the sources in which it exists, such as books, teachers, databases, and scientific instruments. This extracted knowledge must then be integrated together for use by the induction process, in whatever form is appropriate – examples, descriptions, advice, constraints, or anything else. Here we focus on the integration task (extraction involves potentially everything from vision to natural language understanding, and more). Our goal is to begin the process of deriving principles for how knowledge-intensive induction algorithms both do now, and can in the future, provide such integration. The hope is that this will lead to both a useful descriptive framework for organizing existing approaches, as well as a prescriptive framework for generating new approaches that go beyond the existing ones.

We'll make this beginning by presenting two partial frameworks for knowledge integration in induction, along with implications drawn so far by applying them to four recently proposed knowledge-intensive induction algorithms. The focus here is specifically on knowledge integration for induction, rather than on the broader issue of knowledge integration in general, in the hope that the extra structure provided by the induction problem will lead to more powerful integration strategies than have been proposed for the general case. The more developed of the two knowledge-integration frameworks – and thus the one emphasized in this paper – is the *transformational* framework. It describes how knowledge-intensive induction algorithms are, and can be, derived by transforming traditional learning methods. The more tentative *problem-space* framework attempts to go beyond the transformational framework to the more difficult task of characterizing the fundamental components of all induction algorithms, whether knowledge-intensive or not. This framework is covered only briefly here as an intriguing possibility for the future.

THE TRANSFORMATIONAL FRAMEWORK

An induction algorithm can be viewed abstractly as a black box with one output port for the concept description and one or more input ports. A minimal induction algorithm has just one input port, for training data, with all other information being hardwired into the algorithm as a fixed *bias* [6]. The only way such an algorithm can use additional knowledge – other than by reprogramming – is to find some way of recoding the knowledge as pseudo-examples. For example, Quinlan describes how knowledge about type restrictions on the arguments of predicates could conceivably be used indirectly by the FOIL algorithm through the generation of pseudo-negative examples that cause FOIL to eliminate candidate hypotheses that would violate the type restrictions [8].

Most induction algorithms actually do provide some additional input ports that allow explicit provision of other types of information; that is, of knowledge beyond what is embodied in the examples. For example: the candidate elimination algorithm provides an input port for information concerning the partial ordering that defines the initial hypothesis space [4]; FOIL provides an input port for the set of relations that can be used in candidate hypotheses [8]; backpropagation provides input ports for learning-rule parameters, the network structure, and the initial connection weights [11]; and Bayesian learning algorithms provide an input port for prior probabilities [1]. Such ports expand the types of information that can be utilized at induction time, but still provide a very limited means for incorporating the full range of knowledge that may be available.

The transformational framework starts with the basic notion of black boxes and ports, as described above, and views knowledge-intensive induction algorithms as the composition of a core, usually knowledge-lean, algorithm plus a set of transformations. Although not all knowledge-intensive algorithms can be viewed in this fashion, when they can, the results can be quite informative. The four knowledge-intensive algorithms covered in the next section do all fit this framework, and are based on three distinct core algorithms – candidate elimination, FOIL, and backpropagation – and on two general types of transformations to these core algorithms:

- A *reformulation* transformation modifies the core algorithm so that its ports can handle a wider range of inputs, either by generalizing its existing ports or by adding new ones. A simple example of a reformulation is the modification of a decision-tree learner to allow it to accept a task-specific split criterion from an input port, rather than always using the same built-in criterion. A more sophisticated example of a reformulation is IVSM's derivation from the candidate elimination algorithm by converting its examples input port to take a more expressive class of inputs (i.e., version spaces) [3].
- A *preprocessor* transformation adds to the core algorithm a preprocessor that takes a form of input beyond what can be fed directly into the core algorithm's input ports, and translates this broader input into something that one or more of the core input ports can understand.

Quinlan's suggestion of using type constraints to create pseudo-negative examples is exactly a proposal for a preprocessor transformation. The preprocessor would have an input port that can accept type constraints, and would produce negative examples that can be fed into FOIL's existing input port. The combination of FOIL and this preprocessor thus defines a new learning algorithm that can take as input not only examples and relations, but also type constraints.

EXAMPLES

Much recent work on induction algorithms has focused on enhancing their ability to utilize additional knowledge during induction, and thus there are many learning systems we could consider. A full survey of such algorithms is beyond the scope of this paper, so we will focus here on just four recent knowledge-intensive algorithms, each of which provides the ability to utilize EBL-like domain theories, plus possibly some other forms of knowledge:

- IVSM has the ability to utilize EBL-like domain theories plus models of bounded inconsistency [3].
- FOCL has the ability to utilize (possibly partial) EBL-like domain theories plus constraints on predicate arguments [7].
- GRENDEL has the ability to specify the hypothesis space via a formal grammar – which can include an EBL-like domain theory – plus some simple ordering information [2].
- KBANN is a neural network algorithm that has the ability to utilize an EBL-like domain theory [13].

The remainder of this section considers these four systems in more detail.

IVSM is based on the candidate-elimination algorithm (CEA). It is derived by a reformulation of the CEA so that instead of basing its inner loop on the process of updating a version space with respect to a single example, it now updates the version space with respect to a second version space (by intersecting the two version spaces). This reformulation generalizes the CEA's examples input port so that it now accepts version spaces. In addition to this core reformulation transformation, IVSM also uses three distinct preprocessor transformations that are enabled by this reformulated input port. One preprocessor allows IVSM to emulate the CEA by taking examples and converting them into version spaces. A second preprocessor creates version spaces from combinations of examples and EBL-style domain theories. A third preprocessor creates version spaces from combinations of examples and a model of bounded inconsistency. When IVSM is combined with any one of these preprocessors, it actually yields a new induction algorithm: IVSM-CEA, IVSM-EBL, or IVSM-BI.

FOCL is based on FOIL. FOIL uses the information provided on its relations input port to determine what modifications to consider making to the current candidate hypothesis. Essentially, it considers adding the various relations – as instantiated with a mixture of old and new variables – to the current clause of the hypothesis, and uses an information-theoretic measure to determine which possibility is (locally) best. FOCL reformulates this possibility-generation strategy in two ways. First, it increases the set of possibilities by considering adding combinations of relations in a single step, rather than just individual relations. Second, it decreases the set of possibilities by eliminating those that violate constraints on the arguments of the relations (such as type and uniqueness restrictions). The first reformulation supports the addition of an input port for (possibly partial) EBL-like domain theories; the combinations of relations that occur in the condition sides of these rules form the basis for the relation combinations proposed in the reformulated algorithm. The second reformulation supports the addition of an input port for type and uniqueness constraints on the arguments of the relations that are proposed. This new port directly supports knowledge that FOIL would have needed a preprocessor to use.

GRENDEL is also based on FOIL. The core transformation made in developing GRENDEL is also a reformulation of FOIL's generation strategy for possible modifications to the current candidate hypothesis. However, GRENDEL's reformulation is both more radical and more general. GRENDEL generates possibilities by consulting generation rules specified in a context-free grammar. This supports broadening FOIL's input port from one that can take a list of relations to one that can handle a list of context-free grammar rules. A second smaller reformulation allows the processing of possibilities to be selectively deferred, and supports the addition of a second input port to specify this simple ordering information. The remainder of the GRENDEL story is much like IVSM. The generalized input port facilitates the creation of a number of preprocessors that can accept a variety of types of input. This input is then translated into grammar rules that can be fed to this new port. These preprocessors allow GRENDEL to accept the kinds of input utilized by (among others) EBL, FOIL, and FOCL, and thus to emulate these other algorithms. As with IVSM, there is a base GRENDEL algorithm which takes grammars as inputs, and then there are a number of other induction algorithms that are based on GRENDEL, such as GRENDEL-EBL and GRENDEL-FOIL, which are derived from it by adding specific preprocessors.

Finally, KBANN is based on backpropagation. It adds a preprocessor that takes as input an EBL-like domain theory, plus a list of environmental features not covered by the theory, and translates this knowledge into a form that can be fed into backpropagation's initial network topology and initial network weights ports. It leaves backpropagation's remaining input ports – such as its learning-rule parameters – intact.

ANALYZING INPUT PORTS IN THE TRANSFORMATIONAL FRAMEWORK

The transformational framework makes it possible to examine knowledge-intensive learners in more detail, by studying the set of input ports provided by the resulting algorithms, what kind of knowledge they can accept, and what key properties they possess (or fail to possess). Although we are still in the process of identifying what the key properties are for input ports, the list already includes at least two that seem critical.

- The *additivity* of an input port is determined by its ability to accept multiple independent fragments of knowledge at that port. Additivity is important because additive ports can serve directly as integrators for arbitrary amounts of knowledge of the types that they can accept. The prototypical example of an additive input port is the example port in standard induction algorithms. It can accept arbitrary amounts of new information, and combine it straightforwardly with whatever else the system knows. A classical example of a non-additive port is the learning-rate parameter in backpropagation. If more information is available, how should it be combined with what is already known? Must the old information simply be eliminated, and replaced by the new, or should the two values be averaged, or should something else happen?

  For additive ports, the way in which inputs are combined usually depends on their interpretation. Examples can be viewed as constraints on the behavior of the concept being learned, so they are usually combined via an intersection operator. Other types of information might be combined via different operators, such as union or average.

- The *ease of use* of an input port is determined by how easy it is to express knowledge in the language provided by the port. Bayesian priors are a classic case of a difficult-to-use input port, with this difficulty most likely being the single biggest stumbling block in using Bayesian approaches to learning. Sometimes preprocessors can be added to make a port easier to use; however, the port's basic ease of use will still affect how easy it is to write the preprocessors. A good example of such a preprocessor for Bayesian priors is the use of the *minimum description length* principle, which, while it can be viewed as a Bayesian approach, replaces the task of assigning a prior probability to every concept with the arguably simpler

task of choosing an encoding scheme [9].

To illustrate these two properties of knowledge-intensive induction methods, as viewed from the transformational framework, we return to the four algorithms discussed above.

The core IVSM algorithm has two input ports, one for the partial-order information on which the version spaces are based and one for a collection of version spaces. The partial-order port is additive because it can handle an arbitrary number of elements plus ordering relations among them. It is also easy to use, but only for the narrow purpose of identifying (possibly parts of) candidate hypotheses and generality relationships among them. The version-space port is also an additive port – as with the traditional example input port, it can accept an unbounded set of inputs, and combine them (via version-space intersection) with what is already known. Its ease of use is intermediate between that provided by example ports at the low end (at least if they are being used for anything other than just examples) and languages like GRENDEL's grammars at the high end. When IVSM's preprocessors are considered, there are three new input ports, all of which are additive and relatively easy to use (for the restricted uses for which they are intended).

One idea that is directly suggested by this analysis of IVSM is that there is no reason its three distinct preprocessors couldn't all be used simultaneously. Because they all output version spaces, and the version-space port is additive, it should be possible to intermingle information based on examples, domain theories, and bounded inconsistency (thus effectively creating a new algorithm that subsumes the three existing ones).

FOCL's three input ports – for examples, (possibly partial) domain theories, and argument constraints – are all additive, as they can all accept arbitrary amounts of knowledge of their chosen input types. They are also all easy to use for their intended purposes, but difficult to use for other purposes.

GRENDEL's three input ports accept examples, grammars, and ordering information (information about what portions of the hypothesis space should be tried first). Regarding ease of use, the example port has the standard properties; the ordering port is similar to a Bayesian-priors port but likely to be somewhat easier to use because it is much less demanding; and the grammar port is relatively easy to use for most purposes. The example and ordering ports are both additive; however the grammar port is only semi-additive, in that the grammars are closed under union, but not under intersection. Thus the additivity of the grammar port depends on the way in which grammars are used. If a grammar is used as a suggestion as to which hypotheses are most likely – as when grammars are used to encode a domain theory – then grammars can be easily combined with a union operator. However, when grammars are used as constraints on the hypothesis space, it is impossible to generate a separate grammar for each constraint and then integrate the constraints by intersecting the grammars (as IVSM would intersect its version spaces).

KBANN's three input ports accept examples, domain theories, and environmental features. The examples port is much like any other examples port – it is additive and easy to use for its intended purpose (but difficult to use for other purposes). The domain theory port is additive and easy to use. The environmental-features port is like the examples port, being additive and easy to use for very limited purposes.

IMPLICATIONS OF THE TRANSFORMATIONAL FRAMEWORK

Pulling back up now from these detailed analyses to look at the picture more globally, several general implications can be discerned. The first implication is that multiple pieces of knowledge can be combined in three distinct fashions. The first approach feeds the knowledge into multiple of the core algorithm's input ports, and depends on the structure of the core algorithm to perform the integration. For example, KBANN integrates a domain theory with examples by feeding the domain theory to the core

network topology and weight ports, while feeding examples directly to the core examples port. The core algorithm – that is, backpropagation – then combines this knowledge during its normal processing. The second approach utilizes a multi-ported preprocessor that integrates the knowledge provided to its input ports in the process of generating input for the core algorithm. One example is GRENDEL-FOCL's emulation of FOCL via a preprocessor that combines knowledge from all of FOCL's input ports (except for the examples port) in the process of converting this knowledge into a single grammar for use by GRENDEL. A second example is IVSM-EBL's use of a preprocessor to integrate knowledge from its examples and domain-theory ports in the process of generating version spaces for the core IVSM algorithm. The third integration approach is to utilize an additive port that can integrate across multiple pieces of knowledge sent to a single port. IVSM is a good example of this, as its version-space port is an effective additive input port.

The second implication is that the insights underlying different knowledge-intensive algorithms can often be transferred or combined in useful ways. In cases where two knowledge-intensive algorithms are based on the same core algorithm, and where they have transformed the core algorithm in different ways, it should be possible to combine many of the transformations without a great deal of difficulty. For example, GRENDEL's generalization of FOIL's relations port to accept grammars could be combined with FOCL's techniques for pruning hypotheses using typing constraints. It would be an interesting question to see whether this approach would have any advantages over using a preprocessor to incorporate all of FOCL's knowledge into a GRENDEL grammar, as in GRENDEL-FOCL.

In cases where the core algorithms are different, transfer of a more abstract sort can still occur. For example, IVSM's additivity based on version-space intersection leads to asking whether GRENDEL's grammars could support a comparable operation: the answer is no, since context-free grammars are not closed under intersection. This also suggests the new research topic of modifying GRENDEL so that it is more additive. For example, since the intersection of a context-free language and a regular language is a context-free language, it might be possible to create a new version of GRENDEL that has an additive port for regular languages in addition to the existing (non-additive) port for context-free languages.

The third implication is that additional effort would be usefully spent looking at how the two general classes of transformations could be applied to further aspects of existing algorithms, both those considered here as well as others.

BEYOND THE TRANSFORMATIONAL FRAMEWORK

The transformational framework is somewhat unsatisfying for several reasons: it does not apply to all knowledge-intensive learners; it does not apply to knowledge-weak learners (which actually do achieve some forms of knowledge-integration even in simply being able to accept varying numbers of examples and learn from them); and it doesn't say much about how to merge the insights across knowledge-intensive algorithms that have different core algorithms. Our continuing work attempts to go beyond the transformational framework by developing a *problem space* framework that attempts to identify the core functionalities that underly all induction algorithms, and then to understand how all of the knowledge utilized by a learner – examples, domain theories, etc. – is integrated together via its mapping on to these functionalities. In terms of the transformational framework, the goal here can be expressed as finding a single black box and set of input ports that conceptually lie at the heart of all induction algorithms.

The problem-space framework is organized around the concept of the space of candidate hypotheses, thereby continuing the existing line of analyses that have viewed induction as search [12; 5; 10]. In this framework the role of knowledge is first off to specify, constrain, and order the elements – that is, the states – of this space. In the four algorithms we have focused on here, specification of the states in the space occurs rather directly via GRENDEL's grammar port, FOCL's relations port, and IVSM's partial-

order port. Constraints on the set of states considered are provided by IVSM's version spaces and FOCL's argument constraints. Ordering information about the states is provided by GRENDEL's ordering port and GRENDEL's and FOCL's examples ports (though rather indirectly, through their information-theoretic measures). However, none of the four systems allows direct statement of all three types of knowledge. GRENDEL comes the closest, though it requires all constraints to be stated indirectly in terms of what can be generated via the grammar. KBANN is the furthest away, as it cannot accept direct statement of any of these types of knowledge. It does however accept some such information indirectly; for example, its domain theory (plus information about additional domain features) indirectly determines what can and cannot be in the hypothesis space, by determining the network topology.

The remaining use of knowledge in this framework is to provide method-specific knowledge about how to search the space of hypotheses. IVSM is at one extreme, in that it makes no use of such knowledge – it always maintains a representation of all hypotheses that are consistent with all of the knowledge available so far. FOCL, GRENDEL, and KBANN all utilize greedy search algorithms. FOCL uses its relation and domain-theory ports to generate candidate changes at each step, its argument-constraint port to eliminate candidates, and its examples port to order the candidates (via its information-theoretic measure). GRENDEL uses the detailed structure of its grammar rules to generate the candidates at each step – two grammars that generate the same terminal language could lead to different greedy searches if they are specified in terms of different sets of rules. It also uses the information from its ordering port as a first cut at ordering the candidates, and then its examples port to complete the ordering (again via its information-theoretic measure). KBANN uses its examples port to determine the direction in which to descend the gradient in its greedy search (via backpropagation) and its learning-rate port to determine the size of the steps taken in that direction.

Although the problem-space framework is still in a very preliminary stage of development, one insight already revealed by this analysis is that, though all four of the knowledge-intensive algorithms studied here use EBL-like domain theories, they use them in three qualitatively different ways. Two of the algorithms – KBANN and IVSM – trust their domain theories enough to use them to directly affect the space of candidate hypotheses, though they do this in different fashions. KBANN uses the domain theory to specify the initial space of candidate hypotheses (that is, the network structure). In contrast, IVSM uses the domain theory (along with examples) to constrain the space of candidate hypotheses that was earlier generated from information provided to its partial-order port. FOCL distrusts its domain theory sufficiently to allow it to affect only the search strategy; that is, the domain theory is used only to order the search for a hypothesis, and never to prune the space. It thus gets less constraint from its domain theory, but is also able to recover more gracefully if the theory is wrong. GRENDEL's treatment of the domain theory depends on how the domain theory has been converted into a grammar; GRENDEL can employ either a KBANN-like strategy, in which the theory determines the search space, or a FOCL-like strategy, in which the theory orders the search space. In GRENDEL-FOCL, the variant of GRENDEL discussed above, the domain theory orders the search space.

As work continues on the problem-space framework, the insights derived from it should (hopefully) get both broader and deeper.

CONCLUSION

We have begun the process of understanding knowledge-intensive induction algorithms by presenting a transformational framework for creating knowledge-intensive methods from knowledge-weak methods, using the framework to analyze four recent algorithms, and deriving from these analyses general implications about the integration of knowledge in induction. We also described a more preliminary problem-space framework that attempts to identify the core functionalities of any learning method and

how various learning methods are created by mapping out how knowledge sources can be used to define these functionalities.

Beyond what has already been described, one fundamental insight revealed by these two frameworks, and the analyses they yield of existing knowledge-intensive learners, is a path towards simple yet powerful knowledge-intensive induction algorithms. First, additive ports need to be developed that provide broad languages for the basic functionalities of specifying, constraining and ordering hypothesis spaces. Ideally, such ports and languages should combine, for example, the best aspects of IVSM's version spaces and GRENDEL's grammars, yet still cover all of these basic functionalities. Second, comparable ports need to be developed to allow knowledge to be used in whatever search method is chosen. For greedy methods, this tends to be knowledge about proposing, constraining, and ordering the options at each step. Third, a range of preprocessors need to be created that can translate a wide variety of forms of knowledge into these ports. Ultimately this leads to a direct concern about knowledge extraction, as the preprocessors get closer and closer to the prime sources of knowledge (such as books), and thus raises a variety of additional issues about how and when knowledge is extracted. Ultimately the hope is to complete these two frameworks, fuse them into a single more comprehensive framework, analyze the full space of existing knowledge-intensive induction algorithms, and use the resulting insights to build one or more new algorithms that go significantly beyond the existing ones.

ACKNOWLEDGMENTS

This project is partially supported by the National Aeronautics and Space Administration (NASA) under cooperative agreement number NCC 2-538.

REFERENCES

1. Buntine, W. Classifiers: A theoretical and empirical study. 12th International Joint Conference on Artificial Intelligence, Sydney, 1991, pp. 638-644.
2. Cohen, W. W. Compiling prior knowledge into an explicit bias. Machine Learning: Proceedings of the Ninth International Workshop, Aberdeen, 1992, pp. 102-110.
3. Hirsh, H.. *Incremental Version Space Merging: A General Framework for Concept Learning.* Kluwer Academic Publishers, Boston, MA, 1990.
4. Mitchell, T. M. Version spaces: A candidate elimination approach to rule learning. Proceedings of the 5th International Joint Conference on Artificial Intelligence, IJCAII, Cambridge, MA, 1977, pp. 305-310.
5. Mitchell, T. M. An Analysis of Generalization as a Search Problem. Proceedings of IJCAI-79, 1979.
6. Mitchell, T. M. The Need for Biases in Learning Generalizations. Tech. Rept. CBM-TR-117, Department of Computer Science, Rutgers University, 1980. (Reproduced in Shavlik, J. W. & Dietterich, T. G. (1990). *Readings in Machine Learning.* San Mateo, CA: Morgan Kaufmann.)
7. Pazzani, M. J., & Kibler, D. "The utility of knowledge in inductive learning." *Machine Learning 9* (1992), 57-94.
8. Quinlan, J. R. "Learning logical definitions from relations." *Machine Learning 5* (1990), 239-266.
9. Quinlan, J. R. & Rivest, R. L. "Inferring decision trees using the minimum description length principle." *Information and Computation 80* (1989), 227-248.
10. Rosenbloom, P. S. Beyond generalization as search: Towards a unified framework for the acquisition of new knowledge. Proceedings of the AAAI Symposium on Explanation-Based Learning, AAAI, Stanford, CA, 1988, pp. 17-21.
11. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Eds., Bradford Books/MIT Press, Cambridge, MA, 1986, pp. 318-362.
12. Simon, H. A., & Lea, G. Problem solving and rule induction: A unified view. In *Knowledge and Cognition*, L. W. Gregg, Ed.,Erlbaum, Potomac, MD, 1974, pp. 105-127.
13. Towell, G. G., Shavlik, J. W., & Noordewier, M. O. Refinement of approximate domain theories by knowledge-based neural networks. Proceedings of the Eighth National Conference on Artificial Intelligence, AAAI, Boston, MA, 1990, pp. 861-866.

N94-34051

Discovering operating modes in telemetry data from the Shuttle Reaction Control System

Stefanos Manganaris* Doug Fisher*
Dept. of Computer Science
Vanderbilt University

Deepak Kulkarni
Artificial Intelligence Research Branch
NASA Ames Research Center

August 2, 1993

ABSTRACT

This paper addresses the problem of detecting and diagnosing faults in physical systems, for which suitable system models are not available. We propose an architecture that integrates the on-line acquisition and exploitation of monitoring and diagnostic knowledge. The focus of the paper is on the component of the architecture that discovers classes of behaviors with similar characteristics by observing a system in operation. We investigate a characterization of behaviors based on best fitting approximation models. An experimental prototype has been implemented to test it. We present preliminary results in diagnosing faults of the Reaction Control System of the Space Shuttle. The merits and limitations of the approach are identified and directions for future work are set.

1 INTRODUCTION

One of the tasks that operators of complex systems perform is monitoring: the detection of abnormal system behavior. The identification of the cause of an abnormality, or fault diagnosis, is a second one. Researchers in Artificial Intelligence have been trying to automate both tasks.

Traditional monitoring and diagnosis systems are rule-based: an "expert" encodes faults and associated symptoms in rules. Sophisticated rule-based expert systems can draw inferences based on time histories of data and operate in real-time. However, expert systems suffer in many ways. Acquiring and expressing the required knowledge in usable rules is a difficult task. Strong assumptions in the rules make detecting and diagnosing novel faults

*This research has been supported in part by a grant from NASA Ames (NCC 2-645).

difficult. Finally, maintaining a set of rules is expensive and time consuming.

The model-based approach to monitoring and diagnosis attempts to overcome some of these problems by developing inference engines that can reason using models of systems. Expected system behavior is predicted and then compared with the observed one. Diagnostic inferences are guided by any discrepancies. Model-based systems can handle multiple and novel faults, as long as the model chosen is at the right level of abstraction to explain the faults.

The goal of our research is a general framework for monitoring and fault diagnosis that performs well even in cases where we have neither the required expertise to build a rule-based system, nor sufficiently detailed and otherwise suitable models for a model-based system. By observing a system in operation over time, we attempt to discover patterns in its behavior. Machine learning techniques are used to induce and associate behavior patterns and the conditions under which they were observed. In parallel with knowledge acquisition, monitoring and diagnosis are performed based on the knowledge base built so far. We have experimented with aspects of this general approach using data from the Space Shuttle Reaction Control System (RCS).

2 A TRAINABLE MONITORING AND DIAGNOSIS SYSTEM FOR THE RCS

Operators often detect and diagnose faults by observing how quantities of a system change over time. With experience, they identify patterns over the macroscopic, qualitative, features of behavior and their associated normal or abnormal operating conditions. Macroscopic features refer to the shape of the plot of a quantity over time. Simple features are the average value, the average slope, the average noise level, the period of oscillation, and the frequency spectrum.

Given a physical system, we are interested in discovering the following by observing its behavior over time:

- The *operating modes* or *states* of the system and their *behavior characteristics*. A different underlying model governs the behavior of the system in each state. We are not interested in necessarily discovering that model—some characteristics of the behavior implied are sufficient when they differentiate states.
- The *transitions* from one state to another and the *conditions* associated with them. Conditions may refer to operator commands, system talk-back, quantity values and thresholds, or rule-based combinations of these.

The Trainable Monitoring and Diagnosis System (TMDS) integrates this discovery process with monitoring and diagnosis. A high level description of its architecture is shown in Fig. 1. A data acquisition component is the interface with the monitored system. It preprocesses the monitored signals and sends the results to a knowledge acquisition component and a monitoring & diagnosis component; a knowledge-base is maintained by the first and is

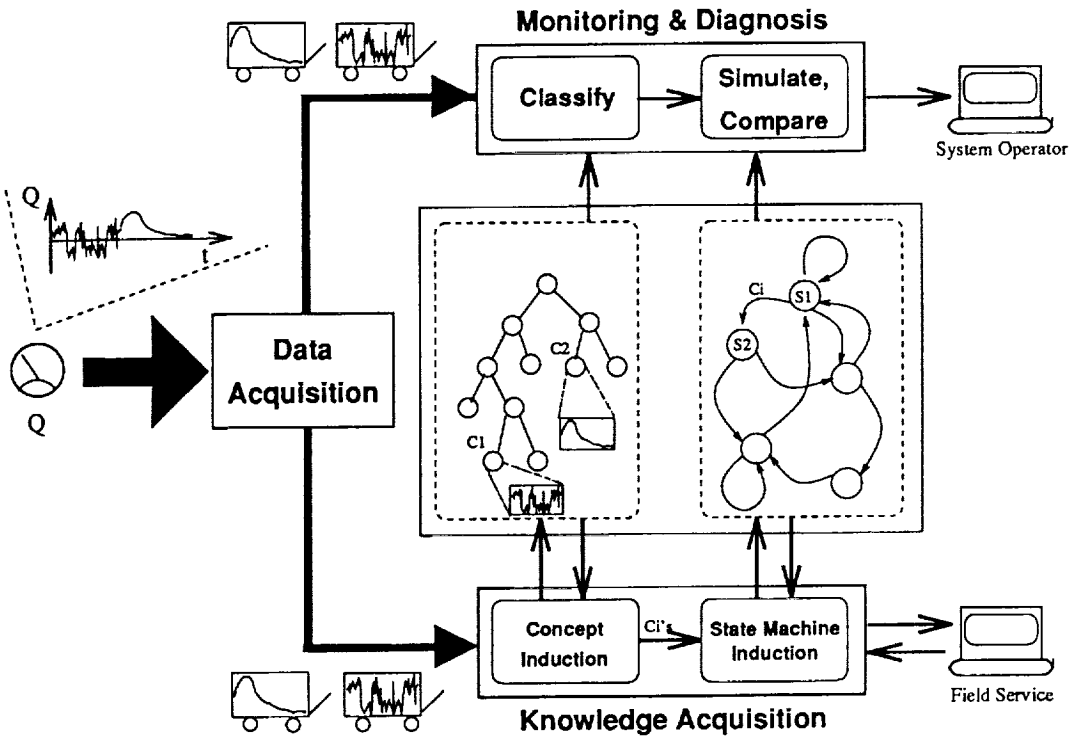


Figure 1: High Level Architecture of the TMDS

used by the second. We demonstrate the current implementation of TMDS with data from the Reaction Control System (RCS) of the Space Shuttle. We experimentally show that behavioral classes discovered by TMDS correspond to a variety of normal and abnormal operating modes of the RCS. We will interleave discussion of general aspects of the TMDS with specific examples in the RCS domain.

The RCS of the Space Shuttle provides propulsive forces from a set of jets to control the motion of the Shuttle (pitch, yaw, roll). It replaces the aerodynamic surfaces, which become ineffective in the upper atmosphere. The RCS is located in three different areas of the orbiter. The forward RCS module is in the fuselage nose area. The aft RCS modules are located in the right and left RCS pods, which are attached to the aft fuselage. Each RCS has two subsystems. One for each propellant: Oxidizer (OX) and Fuel (FU). The OX and FU subsystems are very similar in construction. Each consists of a Helium system, a propellant system (OX or FU), cross-feed and interconnect capabilities, and a jet thruster system. The helium system is used to pressurize the propellant and drive it to the jets. It consists of a Helium tank storing helium under high pressure, two legs in parallel, of two pressure regulators in series, each controlled by a valve. The propellant system consists of a propellant tank and an isolation valve at its output. The jet system consists of a manifold

(pipes and valves), and jets. A FU and an OX pipe goes to each jet. A jet fires when OX and FU are allowed contact in its chamber.

Many quantities of the system are transmitted back to earth via a telemetry link. Each RCS has two He pressure sensors at the He tank, one temperature sensor at the He tank, one pressure sensor for the ullage pressure in the propellant tank, one temperature sensor at the propellant tank, and one pressure sensor at the output of the propellant tank. In addition, every valve's position (talk-back) and every command affecting a valve is also transmitted. More information about the system can be found in the RCS training manual.[3]

3 DATA ACQUISITION IN THE TMDS

Signals monitored can be classified as either analog or digital, depending on whether they exist at every instant of time or not. The data acquisition module of TMDS samples any analog signals and digitizes them.

Further processing partitions the continuous stream of data points into intervals of homogeneous behavior characteristics. After an interval has been identified, the behavior of all signals in that duration is characterized. The result is a stream of *behavior summaries*, one for each interval of system-wide homogeneous behavior. A behavior summary contains characterizations for all monitored signals.

The methods used to characterize signals depend on their types. Signals may be deterministic or random. Deterministic signals can be precisely described by a function of time and are thus predictable; random signals cannot. Deterministic signals may be periodic or aperiodic (transient). Random signals can only be described statistically; we can use approximation models and other methods for deterministic signals.

Our work with the RCS system illustrates how we fleshed out these general issues for a particular system. In this case, TMDS is explicitly instructed how to partition the continuous stream of data points into homogeneous intervals by utilizing commands and talk-back present in the telemetry stream. Commands and verification issued through talk-back indicate when operating modes will change. For each interval discovered in this manner, twelve quantities were monitored in addition to the discrete commands and talk-back for detecting configuration changes. Behavior summaries in this experiment consist of the two parameters of the best fitting *linear* approximation models (i.e., slope and intercept) and the squares of the correlation coefficient for *each* of the twelve quantities, which is a measure of the approximation's 'fitness'. RCS behavior in each interval is thus characterized by thirty-six (3×12) numerical attributes. Linear approximation models were used because they were simple and sufficiently informative for the RCS signals. Although most RCS quantities do not behave linearly, linear approximations were deemed satisfactory for short intervals of time. In RCS operation short behavior summaries are typical.

Returning to general issues, the characteristics of a signal vary depending on the operating mode of the system. The data acquisition module partitions the continuous behavior into homogeneous intervals. The TMDS currently relies on a prespecified subset of the monitored signals to perform this partitioning. Signals that are included in the subset are

known to be indicators of operating mode changes. This was motivated by our application, where, for example, commands to and talk-back from the system are monitored, and they were deemed sufficiently informative. In other applications, more sophisticated techniques may have to be used. For example, partitioning may be based on the detection of abrupt changes in the spectral behavior of a signal,[2] or of abrupt changes in its mean amplitude level.[1] For deterministic signals, for which approximate models are known, the Minimum Message Length principle may be used to decide when a break would yield a "better" description.[15]

4 DISCOVERING OPERATING MODES

We envision a TMDS architecture that exploits a two-part knowledge-base. The first contains knowledge about *system states*. Each state is associated with a class of behaviors of similar characteristics. The second contains knowledge about *transitions* between states and associated conditions. Both states and transitions are annotated. Annotations indicate whether a behavior or a transition is normal or not, and its cause.

Two machine learning components discover and maintain most of the information in the knowledge base. The first one processes behavior summaries as they are generated by the data acquisition module. It discovers classes of similar behaviors. The current TMDS implementation uses COBWEB/3,[10] a portable implementation of COBWEB[6] that handles numeric attributes. The COBWEB system implements an algorithm for data clustering and incremental concept formation. It can be used to organize objects in classes described by a collection of attributes and their values. COBWEB's approach to classification and learning is known as conceptual clustering.[11,8] Learning in COBWEB is unsupervised: no tutor is necessary to provide feedback. We use an unsupervised, conceptual clustering, learning paradigm, because behavior summaries are unlabeled. Labels that correspond to the particular underlying system states (i.e., operating modes) responsible for a behavior are to be *discovered*. COBWEB organizes concepts in a hierarchy, that is, a partial order according to generality. Each concept node of the hierarchy describes a class of behaviors in terms of the same attributes used to characterize behavior summaries. Learning is incremental: COBWEB processes instances one at a time.

The hierarchy evolves by selecting and applying operators that incorporate each instance into the hierarchy. An operator is applied at each level of the hierarchy, starting at the root. At each cluster the algorithm maintains a probability model for the values of each attribute. This information is used by an evaluation function (*category utility*) to select the operator to apply at a particular level for an instance. Category utility prefers operators that result in hierarchies that maximize intra-class similarities and inter-class differences. In particular, for numeric data, COBWEB/3 defines category utility as

$$\frac{\sum_{k=1}^{K} P(C_k) \sum_i 1/\sigma_{ik} - \sum_i 1/\sigma_i}{4K\sqrt{\pi}}$$

where K is the number of clusters, C_k are the individual clusters, σ_{ik} is the standard deviation of attribute i in cluster k, and σ_i is the population-wide standard deviation of attribute

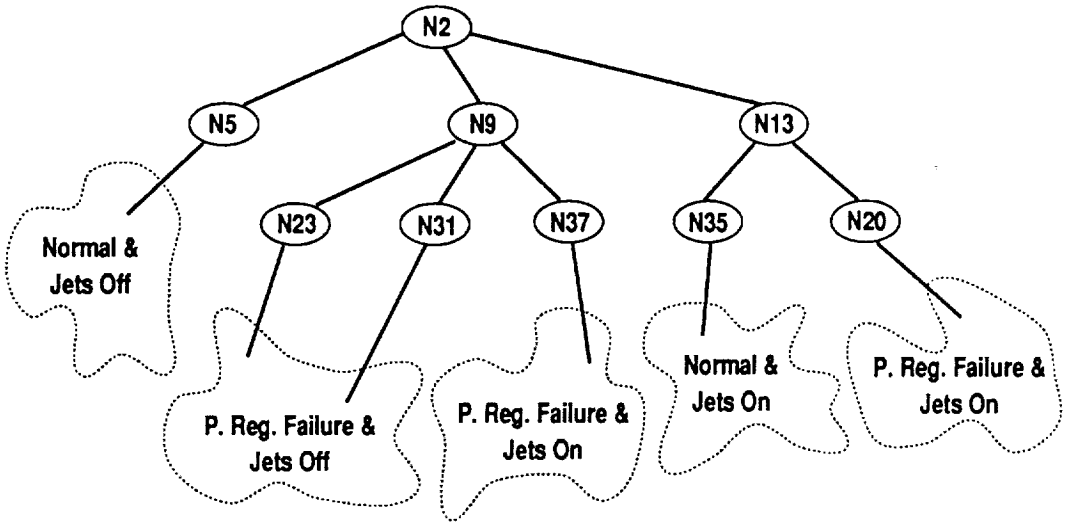


Figure 2: A Hierarchy of Behavior Classes Corresponding to RCS's Operating Modes

i. Intuitively, category utility favors clusters that most reduce the standard deviation over the numeric attributes.

Looking again to the RCS domain, Noisy telemetry data, from several minutes of RCS operation and under various conditions, were processed as described in Section 3 to generate fifty-six behavior summaries. Figure 2 shows a hierarchy of classes formed by COBWEB/3 over these 56 behaviors. The leaf nodes correspond to individual behavior summaries. Clusters were manually labeled according to the configuration of the RCS in the corresponding interval. The operating modes of the RCS we have studied can be roughly classified into four categories: normal, with the jets on or off, and abnormal, with a failed pressure regulator and the jets either on or off. The labels were obviously not used for inducing the classes. One can readily notice that the classes identified are meaningful; for example, node N9 corresponds to the class of behaviors when a pressure regulator has failed, and nodes N35 and N5 correspond to the class of normal behaviors. Figure 3 gives an example cluster definition for a class dominated completely by instances of jets-on, pressure regulator failed-closed behavior. The salient aspects of this definition are circled: Helium ullage in the oxidizer (or fuel) tank and the oxidizer out pressure are dropping, and Helium pressure in the Helium tank is steady.

We envision a second machine learning component that induces a state-machine. Its input is the sequence of states (i.e., concepts) found through categorization with the clustering hierarchy. By observing the state transitions and the changes in operating conditions, a finite automaton is constructed showing possible transitions and conditions associated with them. Related work includes Nordhausen and Langley[13] and Dietterich[5] on sequence induction, and Mitchell et al.[12] in search control learning. This component is not yet operational.

CLASS	PREG-FCL			1.00
JETS	ON			1.00
DURATION	Mean	7.60	SD	2.94
V42P2110C-I	Mean	2968.00	SD	1.00
V42P2110C-S	Mean	0.00	SD	1.00
V42P2110C-R	Mean	0.00	SD	1.00
V42P2112C-I	Mean	2992.00	SD	1.00
V42P2112C-S	Mean	0.00	SD	1.00
V42P2112C-R	Mean	0.00	SD	1.00
V42P2115C-I	Mean	231.32	SD	8.83
V42P2115C-S	Mean	-0.95	SD	1.00
V42P2115C-R	Mean	0.46	SD	1.00
V42P2210C-I	Mean	231.42	SD	8.86
V42P2210C-S	Mean	-0.94	SD	1.00
V42P2210C-R	Mean	0.46	SD	1.00

Figure 3: A cluster definition of jets-on, failed-closed behavior.

5 MONITORING AND DIAGNOSIS WITH THE TMDS

In parallel with the continuous maintenance of the knowledge base, TMDS monitors and diagnoses faults. Behavior summaries, generated by the data acquisition component, are first classified using the classifier constructed by COBWEB. If a behavior summary is very different from all known classes, COBWEB forms a new singleton class. In this way, novel behaviors are detected, and the system operator is warned, even when TMDS has not been trained on them. When a behavior summary is classified to a known faulty class (previously identified by an operator), TMDS can diagnose the fault using information from the annotation of the class. When the classification is to a known normal class, that class is compared to the one predicted by the state machine from the last known state. If the transition at hand is novel, and thus unexpected, or is known to be associated with a malfunction, an appropriate warning may be generated. As we implied earlier, state machine induction is not yet implemented, nor are the diagnostic methods associated with

these structures.

However, to test the accuracy of the acquired knowledge from the current system, we ran experiments, where we predicted the class for behaviors TMDS was not trained on. We focused on behaviors where a pressure regulator has failed closed and on nominal behaviors, under different operating conditions: when different jets fire, for different periods, under different temperatures, etc. For each experiment we trained TMDS on fifty behavior summaries, presented in random order, and tested prediction accuracy on six behavior summaries, which were not used for training. The accuracy achieved was 85.5%, averaged over 30 runs. Given the limited amount of data for training, this level of accuracy is promising. The effects of misclassifications to the overall operation of TMDS are yet to be examined.

6 RELATED WORK AND SUMMARY

This work started at NASA Ames as an alternative approach to monitoring and diagnosis of the RCS. Related work at Ames focuses on a mixed quantitative and qualitative model-based framework for diagnosis over time.[16] Other applications of machine learning for inducing diagnostic knowledge are Pearce's AQR[14] and Lee's and Dvorak's DYNALEARN.[9] Both approaches require a model to systematically simulate system behavior. Pearce focuses on static systems with single faults, and, in a propositional framework, induces a rule that covers a set of behavior examples of the same failure. A single rule is induced for each failure in turn. DYNALEARN extends the ideas of AQR to dynamic systems. It induces a decision tree, which is used to predict suitable starting points in model-based tracking of a time-dependent system.

A simple, yet promising approach to combining induction with model-based reasoning was initiated this Summer by the authors, Peter Robinson of NASA Ames, and Julio Ortega of Vanderbilt University. Our perspective is that models in general, and the RCS models at NASA Ames in particular, can be viewed as superb 'feature extractors' for raw (RCS telemetry) data. In particular, initial experiments used recommendations and intermediate state variables computed by the models to augment the raw (simulated) telemetry data from the RCS system. Decision tree induction was then performed over the augmented data instances. Our goal is to identify those portions of the behavior space in which the models seem to be performing well, which is indicated by the use of model recommendations and state variables in portions of an induced decision tree, and those portions of the behavior space in which the models are not performing well, which are indicated by an absense of model variables/recommendations in other portions of the tree. In the long term, we envision a model construction aid that focuses the attention of the model builder on those portions of the behavior space in which the model can be improved.

Smyth and Mellstrom[17] develop a classifier that can reject classes it was not trained on, using a special class of stochastic models and a supervised learner. This feature is fundamental for the classifier in the TMDS. However, we rely on COBWEB, an *unsupervised* learner, in discovering novel classes.

Nordhausen and Langley[13] address the general problem of empirical discovery. Their

IDS system processes a sequence of temporally ordered qualitative descriptions of states and forms a taxonomy of states, discovers relations between pairs of states (e.g., transitions and conditions on those transitions), and numeric relations of quantities. We anticipate that several ideas of IDS can be used in TMDS.

Previous applications of clustering in diagnosis include.[4] Carnes and Fisher cluster individual snapshots of system behavior to learn fault modes for design (in particular, sensor placement) and diagnosis. Static quantitative models are used to generate training data. Training instances are not temporally related.

In this paper, we have shown preliminary results towards a trainable architecture for monitoring and fault diagnosis. A partial implementation of the TMDS architecture was used successfully to discover the characteristics of the behavior of the Reaction Control System under normal and abnormal operating conditions. Key traits of the TMDS system are the following:

- The TMDS is trainable and can be adapted to monitor and diagnose any system. The characteristics of a system's behavior are discovered by observing it in operation. Novel behaviors are identified as such, and knowledge about their cause is elicited from an operator. Knowledge acquisition is driven by the behavior characteristics of the monitored system—*not* by an "expert". TMDS learns from its experiences. Training continues in parallel with monitoring and diagnosis for reinforcement and refinment of its dynamic knowledge-base.

- Monitoring and diagnostic knowledge is in a compiled form, suitable for real-time performance. Knowledge acquisition is guided by TMDS's discoveries, but is a separable task and can be run off-line.

- Diagnosis is based on system behavior over time, not isolated snapshots. Temporal aspects, which often carry crucial diagnostic information, can be captured in state machines and are used in diagnosis.

- TMDS is expected to be robust with respect to sensor failures. A failed sensor results in either erroneous or no information at all about a signal, which is one focus of cited work by Carnes and Fisher. It has been demonstrated that COBWEB's classifier is robust with respect to noisy or missing attributes.[7] Even with some erroneous values COBWEB can still find a good matching class based on the remaining ones.

Although the proposed approach could be the only choice for new or very complex systems, we believe it could also be the right choice for systems for which a rule- or model-based approach may be used. A rule-based system is bound to be static: its knowledge base can only be modified at great expense. A model-based system's understanding of faults is limited to those that can exist in the model's approximation of reality. A trainable monitoring and diagnosis system would, given sufficient training, be able to handle any fault, subject only to the limitations of the expressiveness of its representation of behaviors.

7 ACKNOWLEDGEMENTS

The first two authors thank the AI branch of the NASA Ames Research Center and Peter Friedland for sponsoring visits at Ames during the past two summers. Special thanks to Pat Langley for contributing important ideas, and to Kevin Thompson for his help with COBWEB/3. Many thanks to Peter Robinson and Matt Barry for providing the telemetry data of the RCS, and to Peter Cheesman and Wray Buntine for suggesting techniques to address some of the issues introduced.

8 REFERENCES

[1] Michele Basseville and Albert Benveniste. Design and comparative study of some sequential jump detection algorithms for digital signals. *IEEE Trans. Acous., Speech, Signal Processing*, ASSP–31:521–534, June 1983.

[2] Michele Basseville and Albert Benveniste. Sequential detection of abrupt changes in spectral characteristics of digital systems. *IEEE Transactions on Information Theory*, IT–29(5):709–724, September 1983.

[3] Richard Bush. *Reaction Control System Training Manual.* NASA, Mission Operations Directorate, Training Division, Flight Training Branch, April 1987. RCS 2102.

[4] Ray Carnes and Douglas H. Fisher. Learning fault modes for design and diagnosis though clustering. Technical Report CS–91–06, Vanderbilt University, Computer Science Dept., Nashville, TN, 1991.

[5] Thomas G. Dietterich. Learning about systems that contain state variables. In *Proc. of the Fourth National Conf. on Artificial Intelligence (AAAI–84)*, pages 96–100, Austin, TX, August 21–26 1984.

[6] Douglas H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[7] Douglas H. Fisher. Noise-tolerant conceptual clustering. In *Proc. of the Eleventh Intl. Joint Conf. on Artificial Intelligence*, pages 825–830, 1989.

[8] Douglas H. Fisher and Pat Langley. Methods of conceptual clustering and their relation to numerical taxonomy. In W. Gale, editor, *Artificial Intelligence and Statistics.* Addison Wesley, 1986.

[9] W. Lee and D. Dvorak. Learning diagnostic knowledge for continuous-variable dynamic systems. In *Working Notes of the Model Based Reasoning Workshop*, pages 129–133, Detroit, MI, 1989.

[10] Kathleen McKusick and Kevin Thompson. COBWEB/3: A portable implementation. Technical Report FIA–90–6–18–2, NASA Ames Research Center, Artificial Intelligence Research Branch, CA, 94035, June 1990.

[11] R. S. Michalski and R. Stepp. Learning from observation: Conceptual clustering. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach.* Morgan Kaufmann, San Mateo, CA, 1983.

[12] T. M. Mitchell, P. E. Utgoff, and R. Banerji. Learning problem solving heuristics by experimentation. In R. S. Michalski, T. M. Mitchell, and J. G. Carbonell, editors, *Machine Learning: An Artificial Intelligence Approach.* Morgan Kaufmann, San Mateo, CA, 1983.

[13] Bernd Nordhausen and Pat Langley. An integrated framework for empirical discovery. *Machine Learning*, 1992. (To appear in the special discovery issue).

[14] D. A. Pearce. The induction of fault diagnosis systems from qualitative models. In *Proc. of the Seventh National Conf. on Artificial Intelligence (AAAI-88)*, pages 353–357, Saint Paul, MN, August 21–26 1988.

[15] R. Bharat Rao and Stephen C-Y. Lu. Learning engineering models with the minimum description length pinciple. In *Proc. of the Tenth National Conf. on Artificial Intelligence*, 1992.

[16] Peter Robinson. Automated fault diagnosis algorithms for the reaction control system of the space shuttle. Technical Report FIA–93–05, NASA Ames Research Center, 1993. (Publication p4).

[17] Padhraic Smyth and Jeff Mellstrom. Detecting novel classes with applications to fault diagnosis. In Derek Sleeman and Peter Edwards, editors, *Proc. Ninth Intl. Conf. Machine Learning*, pages 416–425, 1992.

N94-34052

SELF-CALIBRATING MODELS FOR DYNAMIC MONITORING AND DIAGNOSIS

Benjamin Kuipers
Computer Science Department
University of Texas at Austin
Austin, Texas 78712 USA

August 13, 1993

Abstract

The goal of our work in qualitative reasoning is to develop methods for automatically building qualitative and semi-quantitative models of dynamic systems, and to use them for monitoring and fault diagnosis. Our qualitative approach to modeling provides a guarantee of coverage while our semiquantitative methods support convergence toward a numerical model as observations are accumulated. In recent work, we and our collaborators have developed and applied methods for automatic creation of qualitative models; developed two methods for obtaining tractable results on problems that were previously intractable for qualitative simulation; and developed more powerful methods for learning semi-quantitative models from observations and deriving semi-quantitative predictions from them. With these advances, qualitative reasoning comes significantly closer to realizing its aims as a practical engineering method.

1 Introduction

The world is infinite, continuous, and continually changing over time. Human knowledge and human inference abilities are finite, apparently symbolic, and therefore incomplete. Nonetheless, people normally reason quite effectively about the physical world.

Models of particular systems or mechanisms play an important role in this capability. In service of a task such as diagnosis or design, simulation predicts the behaviors that follow from a particular model. In diagnosis or explanation, these predictions include testable consequences of a diagnostic hypothesis. In design, these predictions make explicit the consequences of a set of design choices.

A qualitative differential equation (QDE) model is a symbolic description expressing a state of incomplete knowledge of the continuous world, and is thus an abstraction of an infinite set of ordinary differential equations models. Qualitative simulation predicts the set of possible behaviors consistent with a QDE model and an initial state. Together, these methods support a meaningful and sound approach to "proof by simulation".

We have developed a substantial foundation of tools for model-based reasoning with incomplete knowledge: QSIM and its extensions for qualitative simulation; Q2, Q3 and their successors for semi-quantitative reasoning on a qualitative framework; and the CC and QPC model compilers for building QSIM QDE models starting from different ontological assumptions.

The QSIM representation for qualitative differential equations (QDEs) and qualitative behaviors was originally motivated by protocol analysis studies of expert explanations. A QDE represents a set of ODEs consistent with natural states of human incomplete knowledge of a physical mechanism. Qualitative simulation can be guaranteed to produce a set of qualitative behavior descriptions covering all possible behaviors of all ODEs covered by the QDE.

The subsequent evolution of QSIM has been dominated by the mathematical problems of retaining this guarantee while producing a tractable set of predictions. A variety of methods now exist for applying a deeper analysis, changing the level of description, or appealing to carefully chosen additional assumptions, to obtain tractable predictions from a wide range of useful models.

Quantitative information can be used to annotate qualitative behaviors, preserving the coverage guarantee while providing stronger predictions. Quantitative information may be expressed as bounds on landmarks and other symbolic elements of the qualitative description [Kuipers & Berleant, 1988], by adaptively inserting new time-points to improve the resolution of the description and converge to a numerical function [Berleant & Kuipers, 1992, 1993], and by deriving envelopes bounding the possible trajectories of the system [Kay & Kuipers, 1992, 1993]. Observations are interpreted by unifying quantitative measurements against the qualitative behavior prediction, yielding either a stronger prediction or a contradiction. As quantitative uncertainty in the QDE and initial state decrease to zero, the resulting behavioral description converges to the true quantitative behavior, though computational costs can still be high with current methods.

We have developed two model-compilers for QDE models: CC, which takes the component-connection view of a mechanism, and QPC, which implements an extended version of Qualitative Process Theory. Other model-compilers for QDEs, e.g. using bond graphs or compartmental models, have been developed elsewhere. These model-building tools will support deeper investigation into selection of views and modeling assumptions.

There are several inference schemes built on the set of all possible behaviors that are particularly well-suited to reliable model-based reasoning for diagnosis and design. For design, desirable and undesirable behaviors can be identified, and additional constraints inferred to guarantee or prevent those behaviors.

For monitoring and diagnosis, plausible hypotheses are unified against observations to strengthen or refute the predicted behaviors. In MIMIC [Dvorak & Kuipers, 1989, 1991], multiple hypothesized models of the system are tracked in parallel in order to reduce the "missing model" problem [Perrow, 1985]. Each model begins as a qualitative model, and is unified with *a priori* quantitative knowledge and with the stream of incoming observational data. When the model/data unification yields a contradiction, the model is refuted. When there is no contradiction, the predictions of the model are progressively strengthened, for use in procedure planning and differential diagnosis. Only under a qualitative level of description can a finite set of models guarantee the complete coverage necessary for this performance.

During the past year, we have made substantial progress in several areas: modeling of complex physical systems; semiquantitative reasoning and monitoring; and tractable qualitative simulation. We also constructed a QSIM model of the Space Shuttle Reaction Control System [Kay, 1992], which serves as a testbed for applying our methods. During the summer of 1992, our group hosted Prof. Lyle Ungar and three of his students from the Chemical Engineering Department at the University of Pennsylvania, who are applying our tools to problems in chemical engineering. The following sections present abstracts of publications summarizing many of our recent results.

2 Automated Model Building

2.1 QPC

Adam Farquhar has built the QPC model compiler into a substantial tool for building domain theories and qualitative models for complex physical systems. Farquhar's doctoral dissertation formalizes and proves the soundness of the QPC model-building algorithm, an essential step toward engineering-quality guarantees.

Automated Modeling of Physical Systems
in the Presence of Incomplete Knowledge

Adam Farquhar
Department of Computer Sciences
University of Texas at Austin

This dissertation presents an approach to automated reasoning about physical systems in the presence of *incomplete knowledge* which supports formal analysis, proof of guarantees, has been fully implemented, and applied to substantial domain modeling problems. Predicting and reasoning about the behavior of physical systems is a difficult and important task that is essential to everyday commonsense reasoning and to complex engineering tasks such as design, monitoring, control, or diagnosis.

A capability for automated modeling and simulation requires

- expressiveness to represent incomplete knowledge,
- algorithms to draw useful inferences about non-trivial systems, and
- precise semantics to support meaningful guarantees of correctness.

In order to clarify the structure of the knowledge required for reasoning about the behavior of physical systems, we distinguish between the *model building* task which builds a model to describe the system, and the *simulation* task which uses the model to generate a description of the possible behaviors of the system.

This dissertation describes QPC, an implemented approach to reasoning about physical systems that builds on the expressiveness of Qualitative Process Theory [Forbus, 1984] and the mathematical rigor of the QSIM qualitative simulation algorithm [Kuipers, 1986].

The semantics of QPC's modeling language are grounded in the mathematics of ordinary differential equations and their solutions. This formalization enables the statement and proof of QPC's correctness. If the domain theory is adequate and the initial description of the system is correct, then the actual behavior of the system must be in the set of possible behaviors QPC predicts.

QPC has been successfully applied to problems in Botany and complex examples drawn from Chemical Engineering, as well as numerous smaller problems. Experience has shown that the modeling language is expressive enough to describe complex domains and that the inference mechanism is powerful enough to predict the behavior of substantial systems.

2.2 QPC Applied to Chemical Engineering

Catino [1993] constructed a large QPC domain theory within chemical engineering for the purpose of doing hazard and operability (HAZOP) studies of moderate-sized chemical process plants. The domain library consists of 50+ model-fragments, and has been used to construct models as large as 280 variables and 340 constraints, making it one of the largest qualitative models ever built. The

abstract of her doctoral dissertation, written under the supervision of Prof. Lyle Ungar, is quoted below.

Automated Modeling of Chemical Plants
with Application to Hazard and Operability Studies

Catherine A. Catino, Ph.D.
Department of Chemical Engineering
University of Pennsylvania

When quantitative knowledge is incomplete or unavailable (e.g. during design), qualitative models can be used to describe the behavior of chemical plants. Qualitative models were developed for several different process units with controllers and recycle, including a nitric acid plant reactor unit, and simulated using QSIM. In general, such systems produce an infinite number of qualitative states. Two new modeling assumptions were introduced, perfect controllers which respond ideally to a disturbance and ignore dynamics in controller variables, and pseudo steady state which ignores transients in all variables. Redundant constraints, reformulated equations, and quantitative information were also used to reduce ambiguity.

A library of general physical and chemical phenomena such as reaction and heat flow was developed in the Qualitative Process Compiler (QPC) representation and used to automatically build qualitative models of chemical plants. The phenomenon definitions in the library specify the conditions required for the phenomena to occur and the equations they contribute to the model. Given a physical description of the equipment and components present, their connectivity and operating conditions, the automatic model builder identifies the phenomena whose preconditions are satisfied and builds a mathematical model consisting of the equations contributed by these active phenomena. Focusing techniques were used to ignore irrelevant aspects of behavior. A dynamic condenser model was automatically generated illustrating QPC's ability to create a new model when a new phase exists.

Based on the ability to automatically build and simulate qualitative process models, a prototype hazard identification system, Qualitative Hazard Identifier (QHI), was developed which works by exhaustively positing possible faults, simulating them, and checking for hazards. A library of general faults such as leaks, broken filters, blocked pipes, and controller failures is matched against the physical description of the plant to determine all specific instances of faults that can occur in the plant. Faults may perturb variables in the original design model, or may require building a new model. Hazards including over-pressure, over-temperature, controller saturation, and explosion were identified in the reactor section of a nitric acid plant using QHI.

3 Tractable Qualitative Simulation

3.1 Qualitative Phase Portraits

The phase portrait is an important representational tool by which engineers capture the possible behaviors of a dynamical system. Wood Wai Lee has just completed a doctoral dissertation in which he shows that qualitative simulation can be used to construct qualitative phase portraits of non-trivial systems, inheriting the QSIM guarantees of complete coverage.

A qualitative method to construct phase portraits
Wood Wai Lee and Benjamin J. Kuipers
AAAI-93

We have developed and implemented a method based on qualitative simulation to construct phase portraits for a significant class of systems of two coupled first order autonomous differential equations, even in the presence of incomplete, qualitative knowledge.

Differential equation models are important for reasoning about physical systems. The field of nonlinear dynamics has introduced the phase portrait representation as a powerful tool for the global analysis of nonlinear differential equations.

QPORTRAIT uses qualitative simulation to generate the set of all possible qualitative behaviors of a system. Constraints on two-dimensional phase portraits from nonlinear dynamics make it possible to identify and classify the asymptotic limits of trajectories and constrain their possible combinations. By exhaustively forming all combinations of features, and filtering out inconsistent combinations, QPORTRAIT is guaranteed to generate all possible qualitative phase portraits.

We have applied QPORTRAIT to obtain tractable results for a number of nontrivial dynamical systems.

Guaranteed coverage of all possible behaviors of incompletely known systems complements the more detailed but approximation-based results of recently-developed methods for intelligently-guided numerical simulation [Nishida et al; Sacks; Yip; Zhao]. Together, these methods contribute to automated understanding of dynamical systems.

3.2 Behavior Abstraction

Daniel Clancy has developed a method for creating a lattice of abstractions of the tree of possible qualitative behaviors, providing a space of alternate descriptions with different degrees of tractability and discriminating power.

Behavior Abstraction for Tractable Simulation
Daniel J. Clancy and Benjamin Kuipers
QR-93

Most qualitative simulation techniques perform simulation at a single level of detail highlighting a fixed set of distinctions. This can lead to intractable branching within the behavioral description. The complexity of the simulation can be reduced by eliminating uninteresting distinctions. Behavior abstraction provides a hierarchy of behavioral descriptions allowing the modeler to select the appropriate level of description highlighting the relevant distinctions. Two abstraction techniques are presented. Behavior aggregation eliminates occurrence branching by providing a hybrid between a behavior tree representation and a history based description. Chatter box abstraction uses attainable envisionment to eliminate intractable branching due to chatter within a behavior tree simulation.

4 Semi-Quantitative Reasoning

Herbert Kay, collaborating with Kuipers and Ungar, has developed two major pieces of the puzzle of semiquantitative simulation. First, he has created, implemented, and proved the soundness of the dynamic envelope method for predicting improved bounds on behavior trajectories, given bounds on landmark values and envelopes around monotonic functions. Second, he and Ungar have developed a new method for learning envelopes around monotonic functions from a stream of observations.

4.1 Predicting Dynamic Bounds on Behaviors

Numerical Behavior Envelopes for Qualitative Models
Herbert Kay and Benjamin Kuipers
AAAI-93

Semiquantitative models combine both qualitative and quantitative knowledge within a single semiquantitative qualitative differential equation (SQDE) representation. With current simulation methods, the quantitative knowledge is not exploited as fully as possible. This paper describes *dynamic envelopes* – a method to exploit quantitative knowledge more fully by deriving and numerically simulating an *extremal system* whose solution is guaranteed to bound all solutions of the SQDE. It is shown that such systems can be determined automatically given the SQDE and an initial condition. As model precision increases, the dynamic envelope bounds become more precise than those derived by other semiquantitative inference methods. We demonstrate the utility of our method by showing how it improves the dynamic monitoring and diagnosis of a vacuum pumpdown system.

4.2 Learning Static Bounds on Functions

Deriving Monotonic Function Envelopes from Observations
Herbert Kay and Lyle H. Ungar
QR-93

Much work in qualitative physics involves constructing models of physical systems using functional descriptions such as "flow monotonically increases with pressure." Semiquantitative methods improve model precision by adding numerical envelopes to these monotonic functions. Ad hoc methods are normally used to determine these envelopes. This paper describes a systematic method for computing a bounding envelope of a multivariate monotonic function given a stream of data. The derived envelope is computed by determining a simultaneous confidence band for a special neural network which is guaranteed to produce only monotonic functions. By composing these envelopes, more complex systems can be simulated using semiquantitative methods.

5 Acknowledgements

This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-8904454, IRI-9017047, and IRI-9216584, and by NASA contracts NCC 2-760 and NAG 9-665. Portions of this paper have appeared in other publications.

6 References

- Daniel Berleant & Benjamin Kuipers. 1993. Qualitative and quantitative simulation: bridging the gap. Submitted for publication.
- D. Berleant & B. Kuipers. 1992. Combined qualitative and numerical simulation with Q3. In Boi Faltings & Peter Struss (Eds.), *Recent Advances in Qualitative Physics*, MIT Press, 1992.
- C. Chiu & B. J. Kuipers. 1992. Comparative analysis and qualitative integral representations. In Boi Faltings and Peter Struss (Eds.), *Recent Advances in Qualitative Physics*, MIT Press, 1992.
- D. Clancy & B. Kuipers. 1993. Behavior abstraction for tractable simulation. In *Working Papers of the Seventh International Workshop on Qualitative Reasoning about Physical Systems*, Orcas Island, Washington.
- D. Dvorak & B. Kuipers. 1989. Model-based monitoring of dynamic systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89).* Los Altos, CA: Morgan Kaufman.
- D. Dvorak & B. Kuipers. 1991. Process monitoring and diagnosis: a model-based approach. *IEEE Expert* 6(3): 67-74, June 1991.
- K. D. Forbus. 1984. Qualitative process theory. *Artificial Intelligence* **24**: 85 - 168.
- P. Fouché & B. Kuipers. 1992. An assessment of current qualitative simulation techniques. In Boi Faltings & Peter Struss (Eds.), *Recent Advances in Qualitative Physics*, MIT Press, 1992.
- Pierre Fouché & Benjamin Kuipers. 1992. Reasoning about energy in qualitative simulation. *IEEE Transactions on Systems, Man, and Cybernetics* **22**(1): 47-63.
- D. W. Franke. Representing and acquiring teleological descriptions. Model-Based Reasoning Workshop, IJCAI-89, Detroit, Michigan, August 1989.
- D. W. Franke. 1991. Deriving and using descriptions of purpose. *IEEE Expert*, April 1991, pp. 41-47.
- D. W. Franke & D. Dvorak. 1989. Component-connection models. Model-Based Reasoning Workshop, IJCAI-89, Detroit, Michigan, August 1989.
- H. Kay. 1992. A qualitative model of the space shuttle reaction control system. University of Texas at Austin, Artificial Intelligence Laboratory Technical Report AI92-188.
- H. Kay & B. Kuipers. 1992. Numerical behavior envelopes for qualitative models. In *Working Papers of the Sixth International Workshop on Qualitative Reasoning about Physical Systems*, Edinburgh, Scotland.
- H. Kay & B. Kuipers. 1993. Numerical behavior envelopes for qualitative simulation. *Proceedings of the National Conference on Artificial Intelligence (AAAI-93)*, AAAI/MIT Press, 1993.

- H. Kay & L. H. Ungar. 1993. Deriving monotonic function envelopes from observations. In *Working Papers of the Seventh International Workshop on Qualitative Reasoning about Physical Systems*, Orcas Island, Washington.
- B. J. Kuipers. 1984. Commonsense reasoning about causality: deriving behavior from structure. *Artificial Intelligence* **24**: 169 - 204.
- B. J. Kuipers. 1986. Qualitative simulation. *Artificial Intelligence* **29**: 289 - 338.
- B. Kuipers. 1987. Abstraction by time-scale in qualitative simulation. *Proceedings of the National Conference on Artificial Intelligence (AAAI-87).* Los Altos, CA: Morgan Kaufman.
- B. Kuipers. 1989. Qualitative reasoning: modeling and simulation with incomplete knowledge. *Automatica* **25**: 571-585.
- B. J. Kuipers. Reasoning with qualitative models. 1993. *Artificial Intelligence* **59**: 125-132.
- B. J. Kuipers. Qualitative simulation: then and now. 1993. *Artificial Intelligence* **59**: 133-140.
- B. J. Kuipers & K. Åström. The composition and validation of heterogeneous control laws. *Automatica*, in press.
- B. Kuipers & D. Berleant. 1988. Using incomplete quantitative knowledge in qualitative reasoning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-88).* Los Altos, CA: Morgan Kaufman.
- B. Kuipers & D. Berleant. 1990. A smooth integration of incomplete quantitative knowledge into qualitative simulation. UT AI RT 90-122.
- B. Kuipers & C. Chiu. 1987. Taming intractible branching in qualitative simulation. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87).* Los Altos, CA: Morgan Kaufman.
- B. J. Kuipers, C. Chiu, D. T. Dalle Molle & D. R. Throop. 1991. Higher-order derivative constraints in qualitative simulation. *Artificial Intelligence* **51**: 343-379.
- B. Kuipers & J. Crawford. 1992. Guaranteed coverage vs intelligent sampling: a reply to Sacks and Doyle. *Computational Intelligence* **8**(2): 289-294.
- B. J. Kuipers & J. P. Kassirer. 1984. Causal reasoning in medicine: analysis of a protocol. *Cognitive Science* **8**: 363 - 385.
- W. W. Lee & B. Kuipers. 1988. Non-intersection of trajectories in qualitative phase space: a global constraint for qualitative simulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-88).* Los Altos, CA: Morgan Kaufman.
- W. W. Lee & B. Kuipers. 1993. A qualitative method to construct phase portraits. *Proceedings of the National Conference on Artificial Intelligence (AAAI-93)*, AAAI/MIT Press, 1993.
- Charles Perrow. 1984. *Normal Accidents: Living With High-Risk Technologies.* New York: Basic Books.
- Bradley L. Richards, Ina Kraan & Benjamin J. Kuipers. Automatic abduction of qualitative models. *Proceedings of the National Conference on Artificial Intelligence (AAAI-92)*, AAAI/MIT Press, 1992.

Session A3: ARTIFICIAL INTELLIGENCE III

Session Chair: Dr. Peter Friedland

N94-34053

FILTERING AS A REASONING-CONTROL STRATEGY: AN EXPERIMENTAL ASSESSMENT

Martha E. Pollack
Department of Computer Science
and
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260

ABSTRACT

In dynamic environments, optimal deliberation about what actions to perform is impossible. Instead, it is sometimes necessary to trade potential decision quality for decision timeliness. One approach to achieving this trade-off is to endow intelligent agents with meta-level strategies that provide them guidance about when to reason (and what to reason about) and when to act. We describe our investigations of a particular meta-level reasoning strategy, *filtering*, in which an agent commits to the goals it has already adopted, and then filters from consideration new options that would conflict with the successful completion of existing goals [1]. To investigate the utility of filtering, we conducted a series of experiments using the Tileworld testbed [12]. Previous experiments conducted by Kinny and Georgeff used an earlier version of the Tileworld to demonstrate the feasibility of filtering [5]. We present results that replicate and extend those of Kinny and Georgeff, and demonstrate some significant environmental influences on the value of filtering.

INTRODUCTION

Many existing and potential AI applications involve systems that are situated in dynamic environments: Laffey *et al.* list examples from aerospace, communications, medical, process control, and robotics applications [6]. Optimal deliberation about what actions to perform is impossible in such environments. This is because all systems have computational resource-limits: their deliberations take time. During the time in which a system in a dynamic environment is deliberating about what actions to perform, the environment may change—and it may change in ways that undermine the assumptions underlying the deliberation. A system may begin a deliberation process with a particular set of available options for action, but new options may arise and formerly existing options may disappear during the course of the deliberation. Moreover, the utilities associated with each option are subject to change during the deliberation. A system that blindly pushes forward with its original deliberation process, without regard to the amount of time it is taking or the changes meanwhile going on, is not likely to make rational decisions about what to do. It is thus sometimes necessary to trade potential decision quality for decision timeliness [14, 10, 13].

One approach is to endow intelligent systems, or agents, with meta-level strategies that provide them guidance about when to reason (and what to reason about) and when to act. In previous work, we have proposed two such strategies: *filtering* [1], and *overloading* [9]. In the present paper, we focus on filtering, a strategy in which an agent commits to the goals it has already adopted, and tends to bypass, or filter from consideration, new options that would conflict with the successful completion of existing goals.

To investigate the utility of filtering, we conducted a series of experiments using a simple, abstract testbed: the Tileworld. Our use of the Tileworld is part of an experimental research methodology that we discuss in detail elsewhere [3, especially Section 5.2]. We first described the Tileworld several years ago [12]. Since then, we have made a number of enhancements to the original system, so that it can support a wider range of experiments. A simplified version of the original Tileworld was used by Kinny and Georgeff [5] in a series of experiments that demonstrated the utility of filtering. The experiments we report on in this paper replicate and extend those of Kinny and Georgeff, and demonstrate some significant environmental influences on the value of filtering.

THE TILEWORLD TESTBED

The Tileworld testbed is a tool that we developed to support controlled experimentation with agents in dynamic environments. It is designed to run under Unix, using Lucid Common Lisp and CLX (the Common Lisp X Interface). We first described the Tileworld several years ago [12]; since then, we have made a number of enhancements to the system, so that it now supports a wider range of experiments. We briefly describe the current state of the system, focusing on those aspects of it that are most pertinent to our experimental investigations of filtering. Details about the system implementation, along with information about how to obtain a copy of it, can be found in the Tileworld User's Guide [4].

The Tileworld consists of an abstract, dynamic, simulated environment with an embedded agent. It is built around the idea of an agent carrying "tiles" around a two-dimensional grid, delivering them to "holes", and avoiding obstacles. The environment is dynamic; during the course of a simulation, objects appear and disappear at rates specified by the researcher. The Tileworld is obviously, and intentionally, a highly artificial environment. In keeping the environment divorced from any particular application, our goal has been to provide a tool that allows researchers concerned with *any* application to focus on what they consider to be key features of that application's environment, without the confounding effects of the actual, complex environment itself. We have, in other words, traded realism—in the short run, at least—for sufficient control to allow for systematic experimentation. This methodological decision is one that has also been made in several other testbeds for studying AI planning, for example, the independently developed NASA Tileworld [8] and the MICE system [2, 7], both of which are also organized around the theme of agents situated

on two-dimensional grids, pushing tiles. See Hanks, Pollack, and Cohen [3] for a discussion of the methodological issues surrounding the use of simplified testbeds.

A researcher using the Tileworld can manipulate and monitor characteristics of the simulated environment (such as how quickly it changes) and of the embedded agent (such as what kind of meta-level reasoning principles it employs). These characteristics can be defined either interactively using a menu-based interface, or by storing parameter settings in files that are then used to control batch-style experiments.

In originally developing the Tileworld, we adopted a minimalist philosophy: our policy was to keep the environment as abstract and simple as possible, in order to provide the experimenter with maximal control over the environment and to ensure that the system's performance is not tied to the particulars of any given domain. Each of the parameters in the original Tileworld was introduced because it represented an abstraction of what we believed to be a potentially important and interesting environmental characteristic. Thus, the original Tileworld allowed us to manipulate a number of environmental characteristics, including the degree of dynamism in the environment, the degree of uniformity of task difficulty, and the degree of uniformity of task reward.

Our early experiences with the Tileworld led us to conclude that, while this was a good set of parameters with which to begin, some extensions were necessary to support the range of experiments we hoped to conduct. In particular, in the original system, agents had only a single type of top-level goal, hole-filling, and no matter how they achieved such a goal, they were always awarded the same score (i.e., the score associated with the hole in question). This made the original Tileworld environment one in which there was very little about which to deliberate, and it was thus difficult to study the trade-offs involved in extra deliberation. We thus extended the system in several ways:

- We added the requirement that agents maintain fuel level: we can thus now study goals of maintenance.

- To enable agents to maintain their fuel levels, we added a "gas station" where they can go to get more fuel. We also added a top-level goal of building stockpiles of tiles having particular shapes at strategic locations on the grid. Thus, where for the original Tileworld agent all top-level goals were of the same type (fill a hole), in the new version there are several different top-level goals.

- We assigned "shapes" to tiles and holes, and changed the reward structure associated with successfully filling a hole. The agent may fill a hole with any tiles, but it gets more points if it uses tiles whose shapes match the shape associated with the hole. As a result, there is now the possibility of investigating trade-offs between the value of alternative plans to achieve a goal. An additional complication is that the agent can carry more than one tile—in the original version it only pushed a tile—but the more tiles it carries, the more rapidly it burns fuel. Again, this means that the quality of alternative alternative solutions to some goal may vary.

In implementing these extensions, we adhered to our original minimalist philosophy: we only introduced those extensions that were needed to support the experiments of interest to us. However, we think that one of the strengths of the Tileworld is its conceptual flexibility: we have found that it is relatively easy to design Tileworld modifications that support experiments that investigate environmental and agent-design issues other than those for which it was originally designed.

EXPERIMENTAL RESULTS

We now present the experiments that we conducted using the Tileworld system, to investigate the properties of filtering as a strategy for controlling reasoning in dynamic environments. Due to space limitations, we do not describe either the motivation or details of the mechanism for filtering here, but see [1, 10, 11]. Our central hypothesis, predicated on the earlier work on IRMA, was that that, in a dynamic environment, a tendency to commit to one's plans can result in overall improved performance, despite the fact that the resulting behavior will sometimes be suboptimal. This hypothesis had previously been explored by Kinny and Georgeff, using a simplified version of the original Tileworld system, along with a somewhat modified notion of filtering [5]. Our first goal was to attempt to replicate the Kinny/Georgeff results in the more-complex environment provided by the enhanced Tileworld system, using the original, better-motivated notion of filtering. We were successful in this: like Kinny and Georgeff, we showed that filtering is an effective control strategy. In addition, we generalized their results: we found that the influence of commitment is bounded, i.e., beyond a certain point, additional commitment does not lead to improved behavior, nor does increased lack of commitment lead to poorer performance. We also observed a relation between the rate of change in the environment and the value of commitment. Here we focus on this final observation.

Our primary experiment used a factorial design with two factors: degree of commitment, for which we had 14 levels, and degree of dynamism, for which we had 11 levels. "Degree of commitment" refers to the strength of the filtering strategy: the most committed agent seldom reconsidered its options until it had completed its current plan, while the least committed agent always interrupted its actions to weigh the significance of perceived changes in the environment. "Degree of dynamism" refers to the average rate of change in the environment: how frequently, on average, do exogenous events occur? The independent parameter was effectiveness, which is a normalized measure of the agent's score. There were a total of 51 trials conducted per experimental condition, where the length of each trial was 80,000 clock ticks. (A clock tick is the amount of time it takes the agent to move one unit of distance in the simulated environment.) Pre-tests were performed to establish the duration of a trial needed to ensure quiescence of effectiveness and to establish the number of trials needed to ensure quiescence of the mean effectiveness across trials.

The data we collected showed that there was a strong tendency for agents that committed more strongly to their plans to achieve higher degrees of effectiveness. This is most strongly evidenced

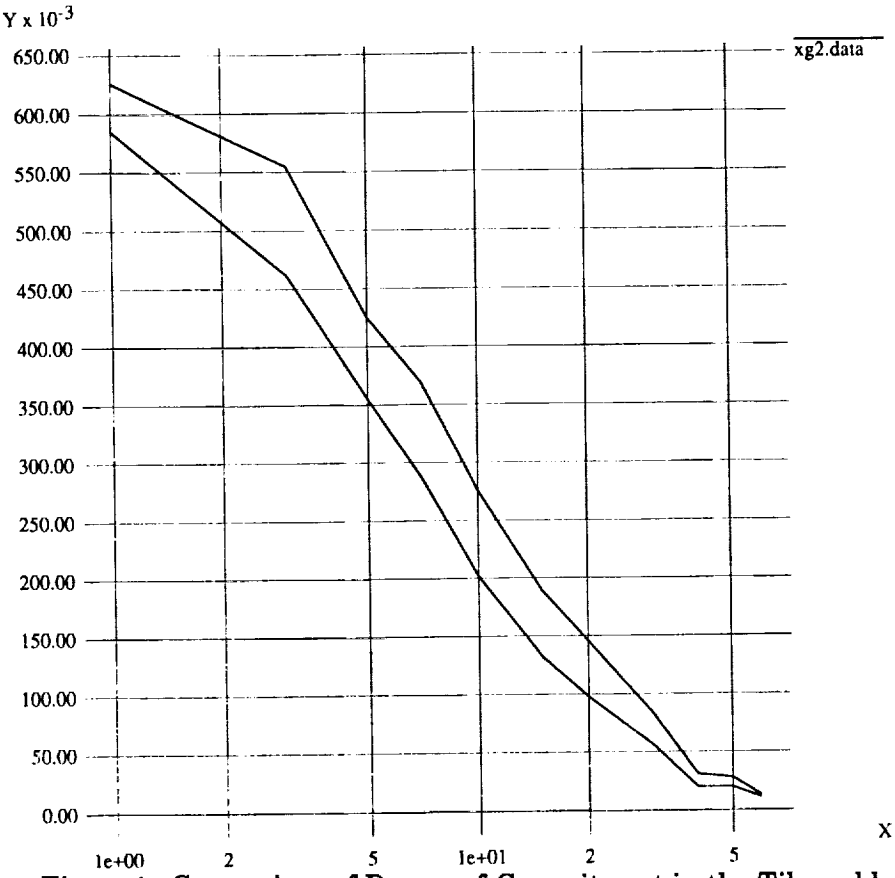


Figure 1: Comparison of Degree of Commitment in the Tileworld

by a comparison of the effectiveness of the most and least committed agents, shown in Figure 1. (When all fourteen levels of committment are plotted, there are some line crossing, but the trend relating effectiveness and degree of commitment is still clear; see [11].)

Table 1 summarizes the significance of the difference in performance between the most committed agent we ran and the least committed one. It shows that the difference between their performance, although not enormous, is statistically significant everywhere except at the endpoints. Further analysis reveals the reason for the collapse at the endpoints. In the slowest environment we studied, there was a great deal of variation in the agent's performance, because it was possible for the environment sometimes to evolve in a way that enabled the agent to succeed at all the tasks it was presented. Because of the high degree of variation in the scores, there was no statistical significance between the agents' performance in these slowly changing environments. At the other endpoint—the most quickly changing environment—the situation is different. In this environment there was very little variation in the scores: both agents scored very poorly, because they were unable to succeed at all but a few of the tasks they were presented. This bottoming effect resulted in a lack of significance between the agents' scores in this environment.

Figure 2 plots the difference in these two agents' performance. The graph shows that the value of commitment, while always positive, is a function of degree of dynamism in the environment. As dynamism increases, the marginal value of commitment first increases, then peaks, and subsequently drops off, although it does not become negative within the bounds of the experiment. This result can be explained as follows. In slower worlds, there are fewer options presented to the agent, and, hence, fewer opportunities for filtering to result in a savings in reasoning cost. Moreover, the advantages of reducing reasoning are minimal, since there is generally enough time to deal with

T-Test Results:
Significance of Difference between Mean Effectiveness
of Most and Least Committed Agents

Dynamism	t	Significance
1	1.121692	$P < .15$
3	3.129425	$P < .0025$
5	3.148238	$P < .0025$
7	4.130018	$P < .0005$
10	5.727610	$P < .0005$
15	6.686329	$P < .0005$
20	7.000076	$P < .0005$
30	4.909135	$P < .0005$
40	3.164884	$P < .0005$
50	2.260098	$P < .02$
60	0.709967	$P < .25$

Table 1: Analysis of Value of Commitment

options. As the world becomes more dynamic, there are more options for consideration, and the penalty for extra reasoning increases, because there is less time to respond to those options. This explains why filtering increasingly pays as dynamism increases. However, another influence comes into play as the rate of change in the environment increases: the missed-opportunity cost grows. As the world changes more rapidly, it becomes increasingly important for the agent to succeed at each individual task, since it will fail to complete a larger proportion of the potential tasks. The shape of the graph in Figure 2 is thus explained by the tension between the increased benefits of reduced reasoning and the increased penalties of missed opportunity, both of which vary directly with rate of change in the world. We expect to see a similar pattern of competing influences on the usefulness of filtering in other domains, and we will pay particular attention to the shape and peak of of the filtering-value curve in other domains, as it reveals useful information about the relative significance of reasoning overhead and missed-opportunity costs.

CONCLUSION

We provided a brief description of a set of experiments aimed at assessing the value of a strategy that may be incorporated in intelligent agents to help focus their reasoning in dynamic environments. The strategy, *filtering*, involves screening from consideration options for action that are incompatible

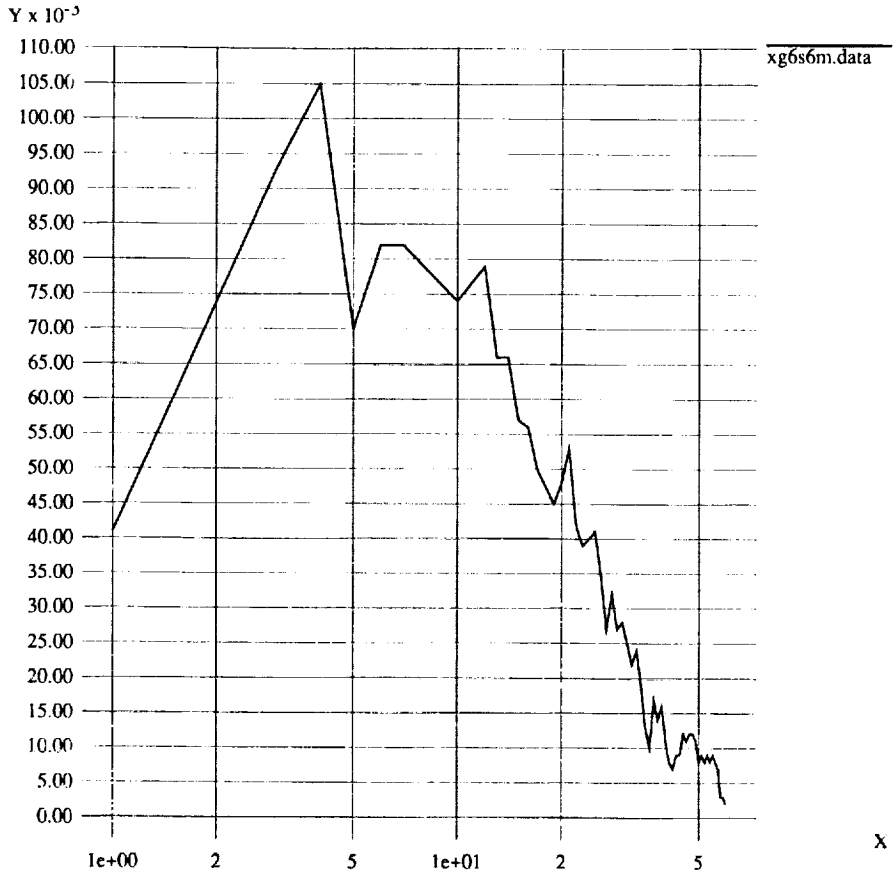


Figure 2: Difference in Effectiveness between Most and Least Committed Agents

with already established plans, except where those options are *prima facie* important enough to trigger a pre-defined override. We relied on a testbed system, the Tileworld, to conduct our experiments. We have made a number of enhancements to the Tileworld since the time it was originally developed, and we described some of the more important of those here. Our experiments demonstrate filtering is a feasible strategy, at least within the Tileworld, a result that suggests to us that it is worth investigating this strategy in more-complex systems. Additionally, our results showed an interesting relationship between the rate of change in the environment and the amount of benefit that one can derive from using a filtering strategy.

Acknowledgements

This work was has been supported been supported by the Air Force Office of Scientific Research (Contracts F49620-91-C-0005 and F49620-92-J-0422), by the Rome Laboratory (RL) of the Air Force Material Command and the Defense Advanced Research Projects Agency (Contract F30602-93-C-0038), and by an NSF Young Investigator's Award (IRI-9258392). Marc Ringuette, Michael Young, Petra Funk, Sigalit Ur, David Joslin, and Arthur Nunes contributed to the development of the Tileworld system and to the experiments described herein.

References

[1] Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988. Also appears in J. Pollock

and R. Cummins, eds., *Philosophy and AI: Essays at the Interface*, MIT Press, Cambridge, MA.

[2] Edward H. Durfee and T. A. Montgomery. MICE: A flexible testbed for intelligent coordination experiments. In Lee Erman, editor, *Intelligent Real-Time Problem Solving: Workshop Report*, pages X1–X16, Palo Alto, CA, 1990. Cimflex Teknowledge Corp.

[3] Steve Hanks, Martha E. Pollack, and Paul R. Cohen. Benchmarks, testbeds, controlled experimentation, and the design of agent architectures. To appear AI Magazine, Winter 1993., 1993.

[4] David Joslin, Arthur Nunes, and Martha E. Pollack. Tileworld users' manual. Technical Report 93-12, University of Pittsburgh Department of Computer Science, 1993.

[5] David N. Kinny and Michael P. Georgeff. Commitment and effectiveness of situated agents. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 82–88, Sydney, Australia, 1991.

[6] T. J. Laffey, P. A. Cox, J.L. Schmidt, S.M. Kao, and J.Y. Read. Real-time knowledge-based systems. *Artificial Intelligence Magazine*, 9:27–45, 1988.

[7] Thomas A. Montgomery and Edward H. Durfee. Using MICE to study intelligent dynamic coordination. In *Second International Conference on Tools for Artificial Intelligence*, pages 438–444. IEEE, 1990.

[8] Andrew B. Philips and John L. Bresina. Nasa tileworld manual. Technical Report FIA-91-11, NASA Ames Research Center, Moffett Field, CA, 1991.

[9] Martha E. Pollack. Overloading intentions for efficient practical reasoning. *Noûs*, 25(4):513–536, 1991.

[10] Martha E. Pollack. The uses of plans. *Artificial Intelligence*, 57:43–68, 1992.

[11] Martha E. Pollack, David Joslin, Arthur Nunes, and Sigalit Ur. Experimental investigation of an agent commitment strategy. In preparation.

[12] Martha E. Pollack and Marc Ringuette. Introducing the Tileworld: Experimentally evaluating agent architectures. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 183–189, Boston, MA, 1990.

[13] Stuart J. Russell and Eric H. Wefald. *Do the Right Thing.* MIT Press, Cambridge, MA, 1991.

[14] Herbert A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69:99–118, 1955.

N94-34054

Translation Between Representation Languages

Jeffrey Van Baalen
Computer Science Department
P.O. Box 3682
University of Wyoming
Laramie, WY 82071
jvb@uwyo.edu

Abstract

A capability for translating between representation languages is critical for effective knowledge base reuse. We describe a translation technology for knowledge representation languages based on the use of an *interlingua* for communicating knowledge. The interlingua-based translation process consists of three major steps: (1) translation from the source language into a subset of the interlingua, (2) translation between subsets of the interlingua, and (3) translation from a subset of the interlingua into the target language. The first translation step into the interlingua can typically be specified in the form of a grammar that describes how each top-level form in the source language translates into the interlingua. We observe that in cases where the source language does not have a declarative semantics, such a grammar is also a specification of a declarative semantics for the language. We describe a methodology for building translators that is currently under development. A "translator shell" based on this methodology is also under development. The shell has been used to build translators for multiple representation languages and those translators have successfully translated non-trivial knowledge bases.

1. Introduction

Acquiring and representing knowledge is the key to building large and powerful AI systems. Unfortunately, knowledge base construction is difficult and time consuming. The development of most systems requires a new knowledge base to be constructed from scratch. As a result, most systems remain small to medium in size. The cost of this duplication of effort has been high and will become prohibitive as attempts are made to build larger systems. A promising approach to removing this barrier to the building of large scale AI systems is to develop techniques for encoding knowledge in a reusable form so that large portions of a knowledge base for a given application can be *assembled* from knowledge repositories and other systems.

For encoded knowledge to be incorporated into a system's knowledge base or interchanged among interoperating systems, the knowledge must either be represented in the receiving system's representation language or be translatable in some practical way into that language. Since an important means of achieving efficiency in application systems is to use specialized representation languages that directly support the knowledge processing requirements of the application, we cannot expect a standard knowledge representation language to emerge that would be used generally in application systems. Thus, we are confronted with a *heterogeneous language problem* whose solution requires a capability for translating encoded knowledge among specialized representation languages.

We are addressing the heterogeneous language problem by developing a translation technology for knowledge representation languages based on the use of an *interlingua* for communicating knowledge among systems. Given such an interlingua, a sending system would translate knowledge from its application-specific representation into the interlingua for communication purposes and a receiving system would translate knowledge from the interlingua into its application-specific representation before use. In addition, the interlingua could be the language in which libraries would provide reusable knowledge bases. An interlingua eases the translation problem in that to communicate knowledge to and from N languages without an interlingua, one must write $(N-1)^2$ translators into and out of the languages. With an interlingua, one need only write 2*N translators into and out of the interlingua.

We consider in this paper the problem of translating declarative knowledge among representation languages using an interlingua with the following properties:

- A formally defined declarative semantics;
- Sufficient expressive power to represent any theory that is representable in the languages for which translators are to be built.

In practice, one cannot expect any given interlingua to have sufficient expressive power to support usable representations of *any* theory that is representable in *any* language. However, an interlingua with the expressive power of first-order logic, such as the Knowledge Interchange Format (KIF) being developed in the ARPA Knowledge Sharing Effort [Genesereth & Fikes 92], can provide that support for a broad spectrum of theories and languages. For our purposes in this paper, we will assume an interlingua and a set of languages for which the properties listed above hold.

The interlingua-based translation process can be thought of as consisting of three major steps:

- Translation from the source language into a subset of the interlingua;
- Translation between subsets of the interlingua; and
- Translation from a subset of the interlingua into the target language.

Since the interlingua is assumed to be at least as expressive as the source language, the first translation step into the interlingua can typically be specified in the form of a grammar that describes how each *top-level form* (e.g., sentence, definition, rule) in the source language translates into the interlingua. Our methodology includes techniques for specifying such grammars so that they are *reversible*, i.e., they can be used not only to translate into the interlingua, but also to translate out of a subset of the interlingua. If one has such a reversible grammar for the target language, then step 2 involves translating from the subset of the interlingua produced by the source language grammar to the subset of the interlingua that is translated (i.e., recognized) by the reverse of the target language grammar. For any given top-level form F_s in the source subset, translation step 2 involves determining a top-level form F_t in the target subset such that F_s is logically equivalent to F_t. Thus, formally, step 2 requires hypothesizing an equivalent form in the target subset and then proving the equivalence.

We have developed the following in support steps 1 and 3:

- A formal description of the translation process into and out of an interlingua;
- A method for determining whether a given grammar in fact specifies how to construct a translation for every top level form in a given source language; and

- A method for determining whether a given grammar is reversible so that it can be used to translate both into and out of an interlingua.

These languages and methods have been incorporated into a "translator shell" system that provides facilities for specifying interlingua-based translation using KIF as the interlingua. The system has been used to build translators for multiple representation languages and those translators have successfully translated non-trivial knowledge bases. Among the systems built so far are a bi-directional CLASSIC [Borgida, et al 89] to KIF translator and a LOOM [MacGregor 91] to KIF translator[Fikes, et al 91].

2. Interlingua-Based Translations and Semantics

We consider here *equivalence preserving translations* [Buvac and Fikes 93] in which the translation of an axiomatization of a logical theory is an axiomatization of an equivalent logical theory. To make such a requirement on translators meaningful, a declarative semantics including logical entailment needs to be formally specified for both the source and target languages. We are assuming such a declarative semantics for the interlingua. In cases where a language does not have such a declarative semantics, specifying a translation of that language into the interlingua provides a declarative semantics for the language. Thus, another advantage of using an interlingua is that it offers a relatively easy way to specify a semantics for new representation languages. This use of an interlingua for specifying the semantics of representation languages may turn out to be at least as important as its role in facilitating translation among representation languages. This method of semantics specification is based on the following definition:

Definition 2.1 (interlingua-based semantics): Let L be a language, L_i be an interlingua language with a formally defined declarative semantics, $\text{TRANS}_{L,Li}$ be a binary relation between top-level forms of L and top-level forms of L_I, and BT_L be a set of top-level forms in L_i. The pair $\langle \text{TRANS}_{L,Li}, BT_L \rangle$ is called an *L_i-based semantics* for L when for every set T_L of top-level forms in L, there is a set T_{Li} of top-level forms in L_i such that

$$\forall s_1 \in T_L \; \exists s_2 \in T_{Li} \; \text{TRANS}_{L,Li}(s_1,s_2)$$

$$\forall s_2 \in T_{Li} \; \exists s_1 \in T_L \; \text{TRANS}_{L,Li}(s_1,s_2)$$

and the theory of $T_{Li} \cup BT_L$ is equivalent to the theory represented by T_L.

Hence, $\text{TRANS}_{L,Li}$ specifies translations of top-level forms in L to top-level forms in L_i. Roughly speaking, BT_L is the set of axioms that are included in the semantics of L expressed in L_i. For example, a device modeling language might have a vocabulary of measures (e.g., INCH, FOOT) and include in its semantics the axioms that relate those measures.

If $\langle \text{TRANS}_{L,Li}, BT_L \rangle$ is being used to define the semantics of L, then "the theory represented by T_L" is equivalent to "the theory of $T_{Li} \cup BT_L$" *by definition*. If L has an independently defined semantics, then the equivalence of the two theories is a requirement on the definition of $\text{TRANS}_{L,Li}$.

TRANS is defined as a relation rather than a function because we allow there to be more than one translation of a top-level form in L so long as it does not matter which translation is picked. Thus, TRANS can be viewed as a function into equivalence classes of interlingua top-level forms. Note also that TRANS defines what it means for two sentences in L to be equivalent, namely that their translations are equivalent sentences in L_i.

An additional advantage of the interlingua-based approach to semantics is that if such a semantics is given in a machine executable form, it can be used to automatically translate a new language into the interlingua. Hence, with a single effort, one can give both a semantics for a new language and a procedure for translating it into the interlingua.

In our language translation methodology one specifies the semantics of a new representation language using a special kind of definite clause grammar [Pereira & Warren 80] that we call a *definite clause translation grammar* (DCTG). This grammar can be used to translate top-level forms in the new language into an interlingua. A DCTG is a set of Horn clauses that has a distinguished binary predicate symbol TRANS such that if s_1 is a top-level form in the new language and s_2 is a top-level form in the interlingua, TRANS(s_1,s_2) follows from the grammar just in case s_2 is a translation of s_1.

We provide a formal technique for showing that such a grammar is a translator, i.e., that for every sentence in the new representation language, the grammar produces a sentence in the interlingua. We also provide a technique for showing that such a grammar is reversible. Both of these techniques have the feature that when a grammar does not have the desired property, they pinpoint locations in the grammar that require repair in order to obtain the property.

3. Translating Between Subsets of the Interlingua

Normally, step 2, translating between subsets of the interlingua, is far more difficult that steps 1 and 3: for each sentence in the source subset of the interlingua we must find an equivalent sentence in target subset, if possible. What makes this difficult is that some sentences have no equivalent sentences in the target subset, while others have such sentences but they are difficult to find.

Our approach to this problem is to treat the target subset of KIF as a *pseudo-canonical* form for KIF and to construct a rewrite system that transforms KIF sentences into this pseudo-canonical form. This use of rewrite systems differs from the standard use [Dershowitz & Jouannaud 90]. Normally one develops a set of rewrite rules from a system of equations that specify equivalences between terms in a language. The goal is to develop a set of directed rules from which it is possible to infer that two terms are equivalent whenever it was possible to infer this from the original undirected equations. An additional goal is to construct rule sets with the following properties: first, given any term t, every possible rewrite sequence from t should end in the same term t'. Second, when two terms are equivalent, rewrite sequences from those terms should end with the same t'. When a set of rules has these properties, we say that every term in the language has a *canonical form* and that the language itself has a *canonical form.*

One can think of the problem of translating into a target subset of KIF as the problem of finding a set of rewrite rules making the target subset a canonical form. Unfortunately, a translator developer does not have a set of equations specifying all the equivalences between terms in KIF and, furthermore, no techniques are known for developing a set of rewrite rules for a particular canonical form. Therefore, we have relaxed some of the requirements on rule sets and call the target subset of KIF a pseudo-canonical form. We provide special rewrite mechanisms that allow a translator to search for rewrite sequences that will lead to sentences in pseudo-canonical form.

4. Status

The KIF-CLASSIC translator was completed in the first three months of the project. In early October 1992, a series of tests of the KIF-CLASSIC translator. The first test translated a "toy" knowledge base from CLASSIC to KIF and then back again. This translation was completely successful, i.e., all of the KIF version of the knowledge base was translated back into CLASSIC. Some of the translations were different than the original CLASSIC statements, however, the resulting knowledge base was equivalent to the original in the sense that CLASSIC did all the same inferences from the translated version as from the original version.

The second test translated into CLASSIC a toy knowledge base that was originally written in KIF. This knowledge base contained knowledge that was appropriate for representation in CLASSIC, however, it was developed by someone who has never used CLASSIC and, hence, the knowledge did not conform to the idioms of the CLASSIC language. Consequently, this KIF knowledge base had a considerably less constrained form and constituted a much more rigorous test of the KIF-CLASSIC translator, requiring it to do many reformulations of the knowledge base in order to get it into a translatable form. Remarkably, this test was also 100% successful in the sense that every statement in the KIF knowledge base was translated into one or more CLASSIC statements.

Having had this much success, it was decided to try a test involving translation from one specialized representation language to another, through KIF. In particular, we translated the ROME Planning Initiative knowledge base from LOOM to KIF using a LOOM-KIF translator developed by Ramesh Patil at USC ISI. Then the KIF-CLASSIC translator was used to translate the result into CLASSIC. One would not expect the translation from KIF-CLASSIC to be 100% successful since LOOM is a strictly more expressive language than CLASSIC.

The first several runs of the KIF-CLASSIC translator translated only around 50% of the KIF knowledge base. However, the translator is designed to flag untranslatable statements and allow the user to assist in their translation. Inspection of the untranslated statements showed that many of them were not correct translations of the LOOM knowledge base into KIF. When these difficulties in the LOOM-KIF translator were repaired, there remained approximately 20% of the KIF version of this knowledge base that the KIF-CLASSIC translator could not translate. Analysis has shown that there is no translation into CLASSIC for this 20% of the KIF knowledge base.

Hence, the KIF-CLASSIC translator succeeded in translating a real LOOM knowledge base into CLASSIC. Every KIF statement generated by the LOOM-KIF translator that was representable in CLASSIC was translated by the KIF-CLASSIC translator. The KIF-CLASSIC translator's ability to flag untranslatable statements proved useful in several ways including debugging the LOOM-KIF translator.

The above tests represent success in all of the milestones planned for this year as well as partially meeting the second milestone planned for next year. Because of this early success, additional unplanned tasks were initiated this year: the development of an EXPRESS to KIF translator and the development of a LOOM-KIF translator. The EXPRESS to KIF translator is currently 95% complete and the LOOM-KIF translator is currently approximately 80% complete.

6. Summary

We have described a methodology for translating knowledge representation languages based on the use of an *interlingua* for communicating knowledge. The interlingua-based

translation process can be thought of as consisting of three major steps: (1) translation from the source language into a subset of the interlingua, (2) translation between subsets of the interlingua, and (3) translation from a subset of the interlingua into the target language. The methodology advocates that the first translation step into the interlingua be specified by a grammar consisting of a set of Horn clauses (called Definite Clause Translation Grammars) that constructively implements a translation predicate relating top-level forms in a source language to their translations in an interlingua. We observed that in cases where the source language does not have a declarative semantics, specifying a translation of that language into the interlingua provides a declarative semantics for the language. Thus, another advantage of using an interlingua is that it offers a relatively easy way to specify a semantics for new representation languages.

A developer of a specialized representation language that desires to build a translator from the specialized language to an interlingua first writes a DCTG G that is an interlingua-based semantics for the language. The developer then uses the methods we have provided to show that G constructs a translation in the interlingua for any top-level form in the specialized language and therefore that G is a translator from the specialized language to the interlingua. The developer then again uses the methods we have provided to show that G also is a translator out of the interlingua in that it constructs a top-level form in the specialized language as a translation for any top-level form in the subset of the interlingua that could be produced by G when it is being used as a translator from the specialized language. Such a reverse translator provides a first approximation of a translator from the interlingua to the specialized language. We provide techniques for augmenting the capability of this first approximation translator. The subset of KIF handled by the reverse grammar is treated as a pseudo-canonical form and the translator developer constructs a rewrite system to transform sentences into this pseudo-canonical form. We provide various methods for assisting with the construction of such a rewrite system.

These languages and methods have been incorporated into a "translator shell" system that provides facilities for specifying interlingua-based translation using KIF as the interlingua. The system has been used to build translators for multiple representation languages and those translators have successfully translated non-trivial knowledge bases.

7. References

[Borgida, et al 89] A. Borgida, R.J. Brachman, D.L. McGuinness, and L.A.Resnick; "CLASSIC: A Structural Data Model for Objects"; Proceedings of the1989 ACM SIGMOD International Conference on Management of Data; Portland, Oregon; May-June, 1989.

[Buvac and Fikes 93] S. Buvac and R. Fikes; "Semantics of Translation"; Proceedings of the workshop on Knowledge Sharing and Information Interchange at the 13th International Joint Conference on Artificial Intelligence; Chambery, France; August 1993.

[Dershowitz & Jouannaud 90] N. Dershowitz and J.-P. Jouannaud; "Rewrite Systems"; in The Handbook of Theoretical Computer Science,;J. van Leeuwen (ed), MIT Press; pp. 245-320, 1990.

[Fikes, et al 91] R. Fikes, M. Cutkosky, T. Gruber, and J. Van Baalen; "Knowledge Sharing Technology Project Overview"; Stanford University Report KSL 91-71, 1991.

[Genesereth & Fikes 92] M. Genesereth and R. Fikes; "Knowledge Interchange Format Version 3.0 Reference Manual"; Logic Group Report Logic-92-1; Computer Science Department, Stanford University; June 92.

[MacGregor 91] R. MacGregor; LOOM Users Manual, USC/Information Sciences Institute; Technical Report Working Paper ISI/WP-22; 1990.

[Pereira & Warren 80] F. C. N. Pereira and D. H. D. Warren; "Definite Clause Grammars for Language Analysis -- A Survey of the Formalism and a Comparison with Augmented Transition Networks, *Artificial Intelligence*, 13, pp. 231-278, 1980.

ACKNOWLEDGEMENTS

Portions of this paper were derived from a paper written jointly by the author and Richard Fikes. Ramesh Patil assisted with the LOOM to KIF grammar. Sasa Buvac and John Cowles participated in helpful discussions. This work is supportedby AFOSR under contract number F49620-92-J-0434, by the Advanced Research Projects Agency, ARPA Order 8607, monitored by NASA Ames Research Center under grant NAG 2-581, and by NASA Ames Research Center under grant NCC 2-537.

N94-34055

A MACHINE-LEARNING APPRENTICE FOR THE COMPLETION OF REPETITIVE FORMS

Leonard A. Hermens
lhermens@eecs.wsu.edu

Jeffrey C. Schlimmer
schlimme@eecs.wsu.edu

School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752

ABSTRACT

Forms of all types are used in businesses and government agencies, and most of them are filled in by hand. Yet much time and effort has been expended to automate form-filling by programming specific systems on computers. The high cost of programmers and other resources prohibits many organizations from benefitting from efficient office automation. A learning apprentice can be used for such repetitious form-filling tasks. In this paper, we establish the need for learning apprentices, describe a framework for such a system, explain the difficulties of form-filling, and present empirical results of a form-filling system used in our department from September 1991 to April 1992. The form-filling apprentice saves up to 87% in keystroke effort and correctly predicts nearly 90% of the values on the form.

INTRODUCTION

Forms are a pervasive part of the operation of modern government and business. As operations become more complex, the forms become increasingly complex too, making it difficult for personnel to complete forms accurately and efficiently. Errors committed by personnel during form-filling can be attributed to general misunderstandings about a particular form or the system in which a form is used. Through the use of machine-learning tools it is possible to assist personnel with repetitious form-filling tasks by providing useful default values for sections of a form, thereby reducing the number of keystrokes necessary to complete a form and reducing the risk of errors. One attractive scenario for automated form processing begins with an office worker who is knowledgeable about a particular task and needs to add information to a form.

Using a personal computer or workstation, a paper form is scanned and transformed into an electronic version. The form appears on a computer screen, with each field on the paper form having a corresponding editable field on-screen. Information may be added to the form while a prediction system assists the user by suggesting default values for blank fields and offers friendly advice about possible inconsistencies in the way the form is filled out (form validation). When the worker has finished with the form, it is sent electronically to others. Again, the computer may offer suggestions to help the user route the form to the appropriate people and track its progress enroute. If desired, the finished form may be printed on a suitable printer. This scenario is within reach of current technology. Scanners of sufficient resolution, computers of sufficient memory and speed, and networking components to link personal computers and workstations are all currently available. Software to enable this scenario, on the other hand, requires three significant components: input, output, and intermediate processing stages. On the input side, researchers are making progress on the problem of assimilating scanned documents [1] and have made considerable progress with the tasks of recognizing the form, segmenting the image into fields, and capturing each field's contents. On the output side, NASA researchers have begun looking at the problems associated with automatically routing forms to the next appropriate worker and validating form content [2]. We focus here on the intermediate processing stage, when the form is actually filled in.

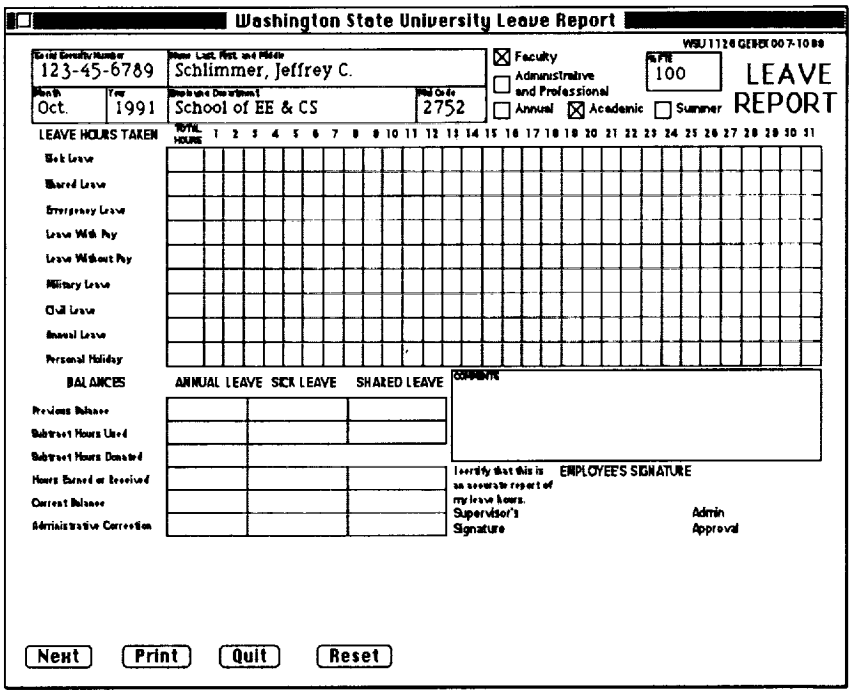


FIGURE 1. The Leave Report form is displayed on-screen in its own window. The user may click the mouse cursor from box-to-box and edit the contents of fields. The control buttons labeled Next, Print, Save, Quit, and Reset are part of the user interface to the form-filling and learning system.

Although a form-filling system can be explicitly programmed for each individual form, there is considerable software engineering overhead for the eventual convenience. Programmers must understand the semantics of the forms in detail, be able to encode specific information into the form-filling program, and then maintain the program as the form itself changes over time. Individuals, companies and government agencies may not have sufficient programmer resources to create and maintain form-filling programs for the hundreds of forms they require to conduct their business. In sharp contrast, programming is not needed with a learning form-filling system because it is able to provide reasonably accurate advice without being explicitly coded to do so. This is one of the hallmarks of such a system.

FORM-FILLING SYSTEM

The focus of this paper is on the intermediate processing stage of form-filling, so we assume the existence of an electronically reproduced, on-screen form—an *electronic form*. The electronic form is created to visually resemble its paper counterpart for two reasons: so users can easily accept the new technology, and so the daily work flow of the user is not adversely affected by the system. Figure 1 shows an electronic version of a Leave Report form used at Washington State University. Although the electronic form appears to be optically scanned image of a paper form, it is actually quite different. Each box, or *field*, on the form is editable; which means a user can type into a text box (e.g., name, address, or social security number), or select a check-box (for selection items) by using the computer's mouse and keyboard. The example form shown in Figure 1 consists of over 300 fields for information input, using both editable text boxes and check-boxes. The user has random access to any of the displayed fields. The control buttons at the bottom of the form labeled Next, Print, Quit, and Reset are part of the user interface to the form-filling and learning system and are not part of a printed Leave Report form.

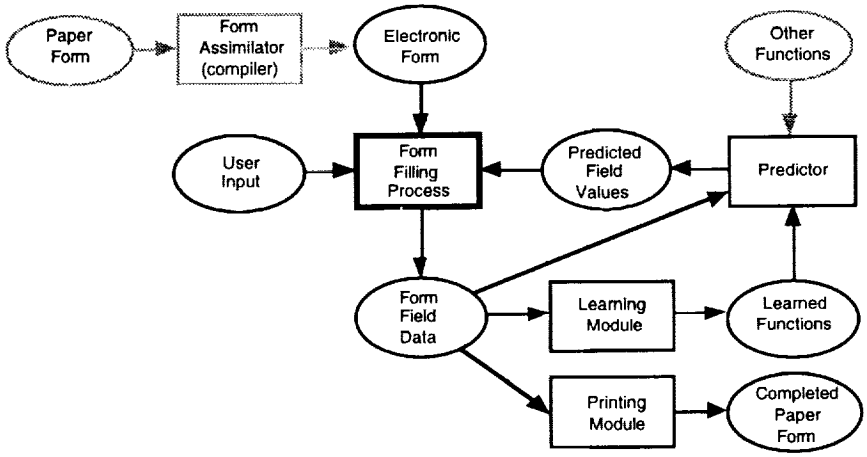


FIGURE 2. A high-level architecture of the form-filling system. Ovals indicate data manipulated by the system, and the boxes are system processes. Arrows indicate the data flow. Although current technology suggests that the automated transformation of a paper form into an electronic one is reasonable and feasible, the shaded oval and box labeled "paper form" and "Form Assimilator (compiler)" were not embodied in the system. The oval titled "Other Functions" indicates expansion possibilities not explored here.

Figure 2 shows a block diagram of the form-filling system used in conjunction with the control buttons. The thick-lined box in the center of the diagram is the core form filling process that combines the electronic form, the user input, and prediction feedback. When a user completes a form field (by typing into an editable box or by clicking a check box), that information is passed as form field data to three modules. Data is presented to the printing module so the user may generate a paper copy of the electronic form. The learning module uses the form data to construct predictive functions; these are used in turn by the predictor module to provide default values for other fields on the form. After each form field is edited by the user, a default value is predicted for each field; after each form is completed, the predictive functions are updated by the learning module.

Although some of the form-filling functions shown in Figure 2 are commonly available in commercial form design packages, our system has an additional component — the learning module. But before we can describe how the learning module plays a role in our system, it is important to first understand how the user interacts with the electronic form.

When the electronic form system is started for the first time, the form fields are blank. To access a field, the user moves the mouse input device to position the screen cursor over a text box or check box. Clicking on a check box will toggle an unchecked box to checked and vice versa. A click on a text box will illuminate a text-edit cursor which indicates that the user may type information into the field. If a field must be changed, the user can employ typical editing commands to delete or change a field's contents. When user has completed the form, they may click on the Print control button to print a paper copy of the form, click Quit to end the session, or click Next begin working on a new instantiation of the same form. Once a new instantiation of the form is shown on the screen, all of the fields are blank again. When a form is being shown on the screen, it is the *current form*. Once the Next button has been clicked, a new instantiation is displayed, and the form that was most recently displayed becomes the *previous form*.

With this model for the completion of electronic forms, we can now discuss the learning component of the system. Suppose a user begins working on form and types values into several fields. The learning module may incorporate any or all of the information for possible future use by the prediction

module. It may use values from fields on the previous form or structure learned from those examples to complete the remaining fields. For example, the system may use the social security number field on the previous form as a predictor for the social security number field on the current form, if it is applicable to do so. If there are no predictions for a field, the field is left blank. If a prediction is made, all applicable fields are updated on the screen. The system will not change any fields that the user has filled in because they are presumed to be confirmed by the user. The form-filling system and the associated prediction methods are very proactive, yet not intrusive. The user does not have to specifically request default predictions because they are always displayed, yet the system is not intrusive in that any default value presented to the user can be easily overridden with normal editing commands.

Using this user interaction model and the architecture shown in Figure 2, a functioning form-filling apprentice program was designed, implemented and used to process 269 Washington State University (WSU) Leave Report forms from September 1991 to April 1992. Although viewed as a prototype, the operational system was fully-functional with respect to form-filling, learning, prediction, and printing. It ran on a Macintosh computer, and was in actual use by three different office support personnel. The system we implemented was tested using each of two learning methods: COBWEB [3] and ID4 [4], an incremental version of ID3 [5].

To increase performance and improve early learning in the system, the learning and prediction subsystems are allowed to use values from fields on the previous form to predict fields on the current form. This means that the system can effectively learn sequences when the user is filling out form in repetition, either sequential or cyclical. For example, for the form shown in Figure 1, after a month of examples (one processing cycle), the system was able to correctly predict the sequence of employees in our department, and the system filled in the appropriate fields on the next copy of the form.

Using COBWEB, field data values from previous forms are used to predict all values of the fields on the current form. In those fields for which there is no prediction, the field is left blank. The prediction cycle is initiated whenever the user clicks the Next button to instantiate a new form. ID4 was slightly different in that it used a bias called *field ranking*. A typical form is designed to be completed from left to right and top to bottom. So each field on the electronic form is assigned an internal numeric rank, increasing first from left to right and then from top to bottom. The prediction mechanism associated with ID4 is prohibited from referencing any field that has a rank higher than the one being predicted on the current form, but is allowed to use example values from the previous forms. There is a learned function for each field and predictions are made independently. The prediction mechanism fires on fields only if the user has not changed the contents of the field, and only if the field's rank is greater than the field being edited. This *form bias* has proven effective, and system responses have been consistent with users' expectations.

EVALUATION

Our system was tested with COBWEB, ID4, and three reference (benchmark) methods including: no-learning (NL), most-commonly-used values (MC), and most-recently-used values (MR). The NL method provided no default values for fields in each new instantiation of the form. This was equivalent to having the user fill out an entire form manually without the aid of a machine learning system. We defined this as the worst-case behavior so that it could be used for comparison with other methods. The MR method predicted the most recent value for a given field, and the MC method predicted the most common. (In case of a tie, the most recent value was predicted.) Form data was collected and saved for each processed form so that a variety of experimental learning methods could be run subsequently to evaluate and compare their performance.

To evaluate the effectiveness of the form-filling apprentice, we used two comparable metrics for the Leave Report form. The first measure was the total number of keystrokes, and the second was the total

number of field prediction errors. A *keystroke error* was recorded for each key that the user typed to override a prediction made by the system or to insert values into an otherwise blank field on the form. *Prediction errors* are measured by counting the number of fields that the user changed, either to delete or insert information. Typing errors introduced by the user were not counted in either of these totals. Results indicate that ID4 reduced the number of required keystrokes by 87% on 269 forms processed, as compared to the no- learning (NL) method. In addition, the prediction-error rate for ID4 was one-tenth that of NL. However, ID4 was dependent on the ordering of the fields on a particular form and was reliant on the order in which the forms were filled out. Performance for COBWEB was not quite as good as ID4 on this task, but it still reduced the number of keystrokes by approximately 64%. When form processing is very cyclical or sequential as in the Leave Report form-filling task, the system was very good at predicting most of the fields for each new form in the sequence using either COBWEB or ID4. We characterize ID4 as accurate, but inflexible, and COBWEB as somewhat flexible with reasonable accuracy.

Figure 3 is a composite graph that shows the percentage of keystrokes saved over no-learning for each learning method. The horizontal, independent axis represents the form processing order from 1 to 269 (labeled chronologically from September to April), and includes two indicators for when new employees were added to the processing cycle. The vertical, dependent axis for each learning method ranges from 0 to 100%. Zero percent corresponds to no correct predictions, 100% indicates all correct predictions, and 50% corresponds to correctly predicting half of the keystrokes. Tick marks to the left of each strip chart show the minimum and maximum keystroke savings in percent, along with the median percentage for the first month and last month of processing. The histogram on the right hand side of each graph shows the percentile density for each method.

Periodic downward spikes in the graph indicate poorer performance in the respective learning methods. This is correlated with either the start of a new month in the cycle or the addition of a new employee to the processing order. All of the methods tested have this characteristic to some degree. ID4 has comparable performance to the other methods during the first month, but then improves dramatically and plateaus for the remaining months. The relatively small number of prediction errors after the month of September can be attributed to two factors: the difficulty of predicting a field value for Previous-Balance-Sick-Leave, and the addition of two new employees to the system in January and March.

Predicting Previous-Balance-Sick-Leave is difficult because the system needs to sum a field value on the current form and a field value on the employee's form from the prior processed month. The dependency between a prior month's form and a current form is very much like connected spreadsheets; a field value in one spreadsheet affects an update on a field on separate but connected spreadsheet. Improved results might be realized when an effective method for learning these spreadsheet-like calculations is developed.

DIFFICULTIES OF FORM-FILLING

As might be expected, some fields are quite easy to predict accurately while others are considerably more difficult. For example, the form in Figure 1 shows five check boxes labeled Faculty, Annual, Administrative/Professional, Academic, and Summer. Although the office support personnel believed that they understood the meaning of these boxes, the semantics were often confusing and resulted in user errors. The machine learning methods we used also had difficulty some of them. For example, Equations 1 and 2 show the desired rules for two of the check boxes on the Leave Report form:

$$\text{Check ACADEMIC when NAME is a faculty member and Month} \in \{Aug, Sep, Oct, Nov, Dec, Jan, Feb, Mar, Apr, May\} \quad \text{(EQ 1)}$$

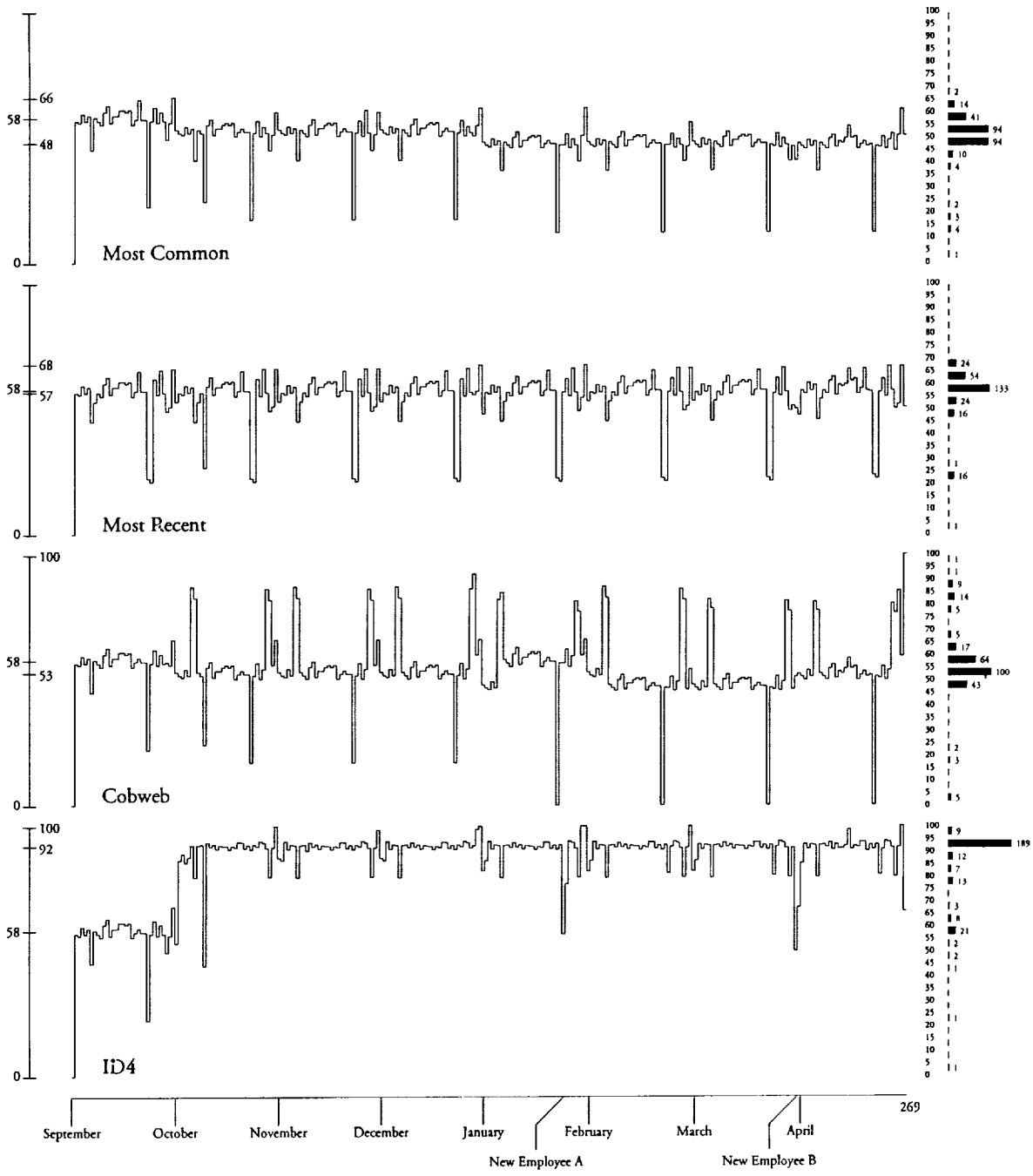


FIGURE 3. A composite graph shows the percentage of keystrokes saved over NL (no-learning) for each learning method tested on the Leave Report form. The horizontal axis is labeled chronologically with indicators for when two new employees were added to the processing cycle. The vertical axis for each learning method indicates the minimum and maximum keystroke savings in percent, as well as the median percentage for the first four months and last four months of processing. A histogram on the right hand side of each graph shows the percentile density for each method.

Check SUMMER when NAME is a faculty member, has summer support, and
Month $\in$ {*May, June, July, August*} (EQ 2)

To be accurate, the system must properly relate a set of months and check boxes to predict the boxes labeled Summer and Academic. One should note that both boxes may be checked in the months of May and August because each of these months are half-summer and half-academic. This subtlety can easily be overlooked by the user, so it is very desirable for the system to accurately predict these fields. Other roadblocks to learning the form shown in Figure 1, for example, include the complex formula for calculating the earned sick leave, which is based on the two above check boxes, plus the %FTE box and a constant value of 8.0. Desired rules for this box are indicated in Equations 3–6:

If Month $\in$ {*Sep, Oct, Nov, Dec, Jan, Feb, Mar, Apr*} and ACADEMIC is checked
then SICK-LEAVE-HOURS-EARNED-OR-RECEIVED is 8.0. (EQ 3)

If Month $\in$ {*May, Aug*} and ACADEMIC is checked, and SUMMER is not checked,
then SICK-LEAVE-HOURS-EARNED-OR-RECEIVED is 4.0. (EQ 4)

If Month $\in$ {*May, Aug*} and ACADEMIC is checked, and SUMMER is checked,
then SICK-LEAVE-HOURS-EARNED-OR-RECEIVED is (%FTE · 8.0) . (EQ 5)

If Month $\in$ {*June, July*} and SUMMER is checked,
then SICK-LEAVE-HOURS-EARNED-OR-RECEIVED is (%FTE · 8.0) . (EQ 6)

There are other rules that can be generated from the two check boxes and the %FTE cells; however, these rules listed above are the only semantically valid conditions for this form. One could imagine that a simple spreadsheet program can handle conditional formulas such as these, but it would require explicit programming by the user. Our desire is to avoid programming systems for complex rules like those shown above, so the goal remains to provide an agent capable of learning rules like these.

RELATED WORK

Our system is somewhat similar to other apprenticeship systems like CAP [6], which was developed to help maintain an appointment calendar. CAP was designed to advise an appointment calendar user in the same way that a knowledgable secretary might. For example, a certain type of meeting may require a certain room at a particular time of day—information that a secretary would know from experience. CAP uses learning from examples to predict three features of newly scheduled appointments: meeting time, duration of meeting, and meeting location. The system has been used to manage a faculty member's appointment calendar.

CAP's user interface is based on the Emacs editor, and the prediction information and queries are presented sequentially to the user. Questions asked of the user are presented using a command-line type dialog, and default prediction values are displayed one-at-a-time. In contrast, our system allows the user to view all of the pertinent information on the active form, on-screen, all at once. This gives the user the advantage of global random access to the form fields and their contents. The user is always in control of the order in which the fields are completed.

CAP is designed to utilize a knowledge base that contains calendar information, a database of personnel information, and other system information like currently active rules, neural network computation data, and a history of user input and commands. Alternatively, our system does not utilize information databases (except for the history of completed form examples), yet it attains reasonable predictions in a relatively short amount of time. A departmental database would aid in the prediction of some fields on the Leave Report form, but empirical results have shown that these fields can be predicted quite well after the first month of training.

CONCLUSION AND FUTURE WORK

We have shown that an apprentice can reduce user effort on repetitive form-filling tasks, and we have described a framework in which these tasks can be accomplished. Our form-filling system yielded reasonable predictions for the fields on our test form. Although the results are promising, the system should be tested over a variety of different forms and typical users. This may reveal broad issues that may not have been uncovered in the confines of a single example form.

Although many issues still abound, we foresee at least two new avenues of research in the future. The first issue is that most paper forms are designed for ease of use within the current paper-oriented workplace—paper forms are not designed for electronic processing. Multiple fragments of information may have been clustered into a single field; a simple example of this is a telephone number field. In the U.S., an area code is a predictor for state of residence. The fact that most forms only have one field for telephone number limits the predictive capacity of the area code fragment. It is important, therefore, to find a mechanism by which the syntax of a particular form can be learned so that over-generalized fields can be appropriately partitioned and so that viable predictions can be made.

A second interesting direction for this research is the idea of allowing users to design and create their own forms. In an interactive drawing system, the user may create editable text boxes by drawing their shape with an input device. Then they may complete their personalized form with the aid of an apprentice system. The system may also suggest alternative representations of the user's form, when appropriate, by adding check boxes or converting fields to selection lists to create a more convenient and useful form. These seemingly independent paths for future research might be easily integrated and provide for additional challenges in learning form-filling.

ACKNOWLEDGMENTS

We would like to thank Susan Wallen, Holly Snyder, and Marie Roberts for making themselves available to use and evaluate our system. We also thank them for their cooperation and patience. We would like to thank the anonymous reviewers who provided several helpful suggestions after reading an earlier draft of this paper. This work was supported in part by the National Science Foundation under grant number 92-1290 and by Digital Equipment Corporation. Computing support was provided by Apple Macintosh Common Lisp, and DeltaPoint's DeltaGraph.

REFERENCES

1. Fisher, J., Hinds, S. and D'Amato, D., "A Rule-based System for Document Image Segmentation," *Proceedings of the Tenth International Conference on Pattern Recognition*, IEEE Press, Atlantic City, NJ, 1990, pp. 567–572.
2. Compton, M., Stewart, H., Tabibzadeh, S., and Hastings, B., "Intelligent Purchase Request System," In *Tech. Rep. No. FIA-92-07*, NASA Ames Research Center, Moffett Field, CA, 1992, pp. 56–57.
3. Fisher, D. H., "Knowledge Acquisition via Incremental Conceptual Clustering," *Machine Learning*, Vol. 2, No. 2, 1987, pp. 139–172.
4. Schlimmer, J.C. and Fisher, D., "A Case Study of Incremental Concept Induction," *Proceedings of the Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, AAAI Press, pp. 496–501.
5. Quinlan, J.R., "Induction of Decision Trees," *Machine Learning*, Vol. 1, No. 1, 1986, pp. 81–106.
6. Dent, L., Boticario, J, McDermott, J., Mitchell, T. and Zabowski, D., "A Personal Learning Apprentice," *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI Press, San Jose, CA, 1992, pp. 96–103.

N94-34056

Automated Knowledge-Base Refinement

Raymond J. Mooney
Department of Computer Sciences
University of Texas
Austin, TX 78712
mooney@cs.utexas.edu

Abstract

Over the last several years, we have developed several systems for automatically refining incomplete and incorrect knowledge bases. These systems are given an imperfect rule base and a set of training examples and minimally modify the knowledge base to make it consistent with the examples. One of our most recent systems, FORTE, revises first-order Horn-clause knowledge bases. This system can be viewed as automatically debugging Prolog programs based on examples of correct and incorrect I/O pairs. In fact, we have already used the system to debug simple Prolog programs written by students in a programming languages course. FORTE has also been used to automatically induce and revise qualitative models of several continuous dynamic devices from qualitative behavior traces. For example, it has been used to induce and revise a qualitative model of a portion of the Reaction Control System (RCS) of the NASA Space Shuttle. By fitting a correct model of this portion of the RCS to simulated qualitative data from a faulty system, FORTE was also able to correctly diagnose simple faults in this system.

1 Introduction

The problem of revising an imperfect knowledge base (domain theory) to make it consistent with empirical data is a difficult problem that has important applications in the development of expert systems (Ginsberg et al., 1988). Knowledge-base construction can be greatly facilitated by using a set of training cases to automatically refine an imperfect, initial knowledge base obtained from a text book or by interviewing an expert. The advantage of a refinement approach to knowledge-acquisition as opposed to a purely empirical learning approach is two-fold. First, by starting with an approximately-correct theory, a refinement system should be able to achieve high-performance with significantly fewer training examples. Therefore, in domains in which training examples are scarce or in which a rough theory is easily available, the refinement approach has a distinct advantage. Second, theory refinement results in a structured knowledge-base that maintains the intermediate terms and explanatory structure of the original theory. Empirical learning, on the other hand, results in a decision tree or disjunctive-normal-form (DNF) expression with no intermediate terms or explanatory structure. Therefore, a knowledge-base formed by theory refinement is much more suitable for supplying meaningful explanations for its conclusions, an important aspect of the usability of an expert system.

Over the past five years, we have developed a series of machine learning systems that automatically revise incomplete and incorrect domain theories. Section 2 briefly reviews five systems that we have developed and summarizes results from each of them. Section 3 discusses one of these systems, FORTE, in a little more detail. FORTE revises first-order Horn-clause theories and can be viewed as automatically debugging Prolog programs based on examples of correct and incorrect I/O pairs. We briefly summarize our results with using FORTE to debug simple Prolog programs written by undergraduate students and to induce, revise, and diagnose a qualitative model of a portion of the Space Shuttle Reaction Control System.

2 A Series of Knowledge-Base Refinement Systems

By integrating ideas from both explanation-based learning and inductive learning, my students and I have developed a series of systems for automatically revising imperfect knowledge bases of increasing representational complexity.

First, we developed a method called IOU (Induction Over the Unexplained) (Mooney, 1993) for refining overly-general, propositional, Horn-clause domain theories (i.e. if-then rule bases without variables). IOU uses explanation-based methods to learn part of a concept and uses inductive methods over unexplained aspects of examples to impose additional constraints on the final definition. Experiments on real-world data sets for diagnosis of soybean diseases (Michalski and Chilausky, 1980) and human hearing disorders (Porter et al., 1990) demonstrated IOU's ability to use incomplete theories to learn more accurate concepts from fewer examples than a purely inductive learning method like ID3 (Quinlan, 1986). Results in learnability theory (Ehrenfeucht et al., 1989) where used to prove that, under certain conditions, IOU is guaranteed to learn a PAC (probably approximately correct) concept from fewer examples. Finally, IOU was used to model some recent psychological data demonstrating the effect of background theories on human concept acquisition (Wisniewski, 1989). Unfortunately, IOU was restricted to repairing only a certain type of overly-general theory.

Our next system, EITHER (Explanation-based and Inductive Theory Extension and Revision) (Ourston and Mooney, 1990; Ourston and Mooney, in press; Ourston, 1991) was able to refine arbitrarily incorrect propositional Horn-clause theories. EITHER used generic components for deduction, abduction, and induction (ID3) to learn new rules, delete incorrect rules, add antecedents to existing rules, and remove existing antecedents. EITHER was able to successfully refine real expert rule-bases for recognizing promoters in DNA sequences (Towell et al., 1990) and diagnosing soybean diseases, improving the classification accuracy of both theories 30 percentage points using 100 training examples.

Our third system, FORTE (First-Order Revision of Theories from Examples) (Richards and Mooney, 1991; Richards, 1992) was able to refine *first-order* Horn-clause theories by incorporating recently developed methods in inductive logic programming (Muggleton, 1992). FORTE can be viewed as automatically debugging Prolog programs based on examples of correct and incorrect I/O pairs. In fact, it was successfully used to debug simple Prolog programs written by students in an undergraduate course on programming languages. FORTE was also used to automatically induce and revise qualitative models of several continuous dynamic devices from qualitative behavior traces. In

particular, the system induced and revised a qualitative model of a portion of the Reaction Control System (RCS) of the NASA Space Shuttle. FORTE is discussed in more detail in the next section.

Our fourth theory refinement system, RAPTURE (Revising Approximate Probabilistic Theories Using a Repository of Examples) (Mahoney and Mooney, 1993; Mahoney and Mooney, in press) combines symbolic and neural-network learning to refine a certainty-factor rule base. Therefore, this project extended our methods to knowledge bases involving uncertain reasoning. RAPTURE converts a certainty-factor rule base into a network and uses a modified version of connectionist backpropagation (Rumelhart et al., 1986) to adjust certainty factors. If adjusting certainty-factors is insufficient, a symbolic method based on ID3's information gain metric is used to add new rules. Backpropagation and rule addition continue in a cycle until all of the training examples are classified correctly. RAPTURE has successfully revised knowledge bases for three real-world problems: DNA promoter recognition, soybean diagnosis, and the diagnosis of bacterial infections (a version of the MYCIN rule base from Ma and Wilkins (1991)). On the promoter problem, RAPTURE performs significantly better than our previous system, EITHER, and produces a simpler and slightly more accurate rule base than KBANN (Towell et al., 1990), a more standard neural-network theory revisor.

Our most recent system, NEITHER (New EITHER) (Baffes and Mooney, 1993) is a much faster, redesigned version of EITHER that can revise theories with "M of N" rules (rules that fire if any subset of size at least M of their N antecedents are satisfied). It has been tested on refining the promoter domain theory, producing an even simpler rule base with accuracy similar to that produced by RAPTURE and KBANN.

Figure 1 shows learning curves demonstrating the performance of various systems on the DNA promoter recognition problem. A promoter is a genetic region that initiates the first step in the expression of an adjacent gene (*transcription*) by RNA polymerase. The input features are 57 sequential DNA nucleotides (with values A, G, T or C). The data contains 106 examples evenly split between positive and negative. The expert theory provided with the data set, which has 11 rules with a total of 76 literals, is completely overly-specific (proves none of the examples are promoters) and therefore has an initial classification accuracy of only 50%.

The learning curves were generated as follows. Each data set was divided into training and test sets. Training sets were further divided into subsets, so that the algorithms could be evaluated with varying amounts of training data. After training, each system's accuracy was recorded on the test set. To reduce statistical fluctuations, the results of this process of dividing the examples, training, and testing were averaged over 25 runs. Results are shown for the theory revision systems EITHER, RAPTURE, NEITHER (without "M of N" revisions), NEITHER-M-OF-N (with "M of N" revisions), and KBANN, and for the purely inductive systems ID3 (Quinlan, 1986) and neural-network backpropagation (Rumelhart et al., 1986).

All of the revisions systems greatly improve the accuracy of the initial knowledge base and generally perform better than pure induction. The strict rule-based systems (EITHER, NEITHER, and ID3) perform relatively poorly since some aspects of the promoter concept are knowm to fit an M-of-N format. There are several potential sites where hydrogen bonds can form between the DNA and a protein and if enough of these bonds form, promoter activity can occur. Connectionist, probabilistic, and explicit M-of-N systems can represent such concepts more easily than strict Horn-clause theories, which require "M choose N" separate rules to represent an M-of-N concept.

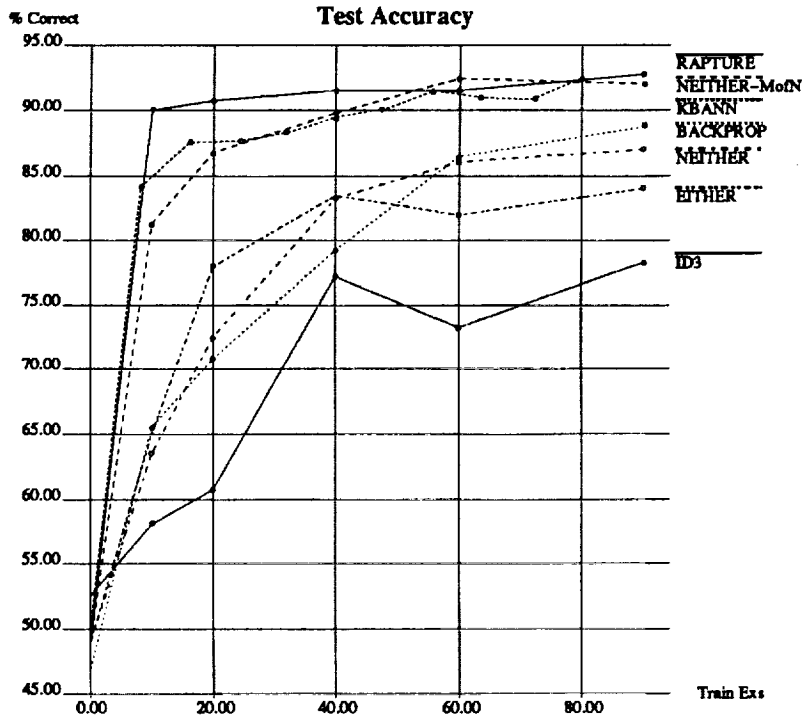


Figure 1: Learning Curves for DNA Promoter Data

3 Overview of FORTE

This section provides a little more detail on our first-order theory revision system. FORTE works by performing a hill-climbing search through a space of specializing and generalizing operators in an attempt to find a minimal revision to a theory that makes it consistent with a batch of training examples.

First, FORTE attempts to prove all positive and negative instances in the training set using the current theory. Positive (negative) instances are tuples of constants that should (should not) satisfy the goal predicate. When a positive instance is unprovable, some program clause needs to be generalized. All clauses that failed during the attempted proof are candidates for generalization. When a negative instance is provable, some program clause needs to be specialized. All clauses that participated in the successful proof are candidates for specialization.

When an error is detected, FORTE identifies all clauses that are candidates for revision. The core of the system consists of a set of operators that generalize or specialize a clause to correctly classify a set of examples. Based on the error, all relevant operators are applied to each candidate clause. The best revision, as determined by classification accuracy on the complete training set, is implemented. This process iterates until the theory is consistent with the training set or until FORTE is caught in a local maximum, i.e. none of the proposed revisions improve overall accuracy.

FORTE's specialization operators include deleting rules and adding antecedents. Several meth-

Program	# of Programs	Training Set Size	Mean Revision Time	% Correct
directed path	4	121 instances	87 seconds	100%
insert after	9	35 instances	82 seconds	100%
merge sort	10	60 instances	199 seconds	100%

Table 1: Summary of program debugging results.

ods are used to determine appropriate antecedents to add to an overly-general clause. One is a hill-climbing method based on the FOIL system Quinlan (1990) for inducing first-order, function-free, Horn-clause rules using an search heuristic based on information gain. Another is called *relational pathfinding* (Richards and Mooney, 1992) and adds a sequence of literals that form a relational path linking all of the arguments of the goal predicate. Since it adds multiple literals at once, relational pathfinding helps overcome local minima problems in FOIL.

FORTE's generalization operators include deleting antecedents and adding rules. Antecedents are chosen for deletion using a greedy algorithm that attempts to maximize the number of additional provable positive examples without causing additional provable negatives. New rules are learned using FOIL and relational pathfinding. FORTE also includes two additional generalization operators (identification and absorption) based on *inverse resolution* as introduced in Muggleton and Buntine (1988). These operators introduce new rules based on repeated patterns of literals found in existing rules.

3.1 Debugging Student Programs

In order to test FORTE's logic program debugging capabilities, we asked students in an undergraduate class on programming languages to hand in their first attempts at writing simple Prolog programs. They gave us their programs after they had satisfied themselves on paper that the programs were correct, but before they tried to run them. The student programs were distributed among three problems: find a path through a directed graph, insert an element into a list, and merge-sort a list. We collected 23 distinctly different incorrect programs, representing a wide variety of errors ranging from simple typographical mistakes to complete misunderstandings of recursion. FORTE was able to debug all of these programs (see Table 1).

3.2 Qualitative Modelling of the Space Shuttle RCS

FORTE has also been used to induce, revise, and diagnosis qualitative models of continuous dynamic systems. Qualitative models suitable for the QSIM qualitative simulation system (Kuipers, 1986) can be represented as Prolog rules by including an antecedent for each of the constraints in the model (such as the flow out of a tank is a monotonically increasing function of the amount in the tank). FORTE can then use qualitative behaviors of the system as examples to revise such a model.

We have applied this approach to qualitative modelling of the Reaction Control System (RCS) of the NASA Space Shuttle. The RCS consists of a number of identical, parallel components; our test domain consisted of one of these components with its valves in fixed positions. Although space prevents us from giving a complete description of the RCS, a simplified view would contain of three

interconnected tanks, plus the thruster outlet. The first tank contains Helium, which it provides at a constant pressure to the fuel tank. The Helium forces fuel out of the fuel tank and into the manifold. From the manifold, the fuel enters the thruster and ignites to provide thrust.

For the purposes of this section, we assume that the valve leading to the thruster is closed (i.e., the thruster is off), the Helium regulator valve is open and providing a constant-pressure supply of Helium, and the valve between the fuel tank and the manifold has just been opened. If the initial pressure in the manifold is lower than the initial pressure in the fuel tank (so that the system is not immediately at equilibrium), then the fuel flows from the fuel tank into the manifold. Providing this single behavior to FORTE allowed FORTE to induce a model for the RCS equivalent to that produced by a QSIM expert (Kay, 1992).

However, since FORTE is a theory refinement system, we can use it in a more sophisticated way. Suppose that the user has a correct system model, but that the system is behaving incorrectly. In this case, we can use theory refinement to revise the correct system model to reflect the actual system behavior. The resulting changes in the model can be viewed as a diagnosis. One of the failures that can occur in the RCS is a leak in one of the manifolds leading from the fuel tank. In order to isolate the leak, the astronauts shut the valve leading from the fuel tank into the manifolds. They then isolate the suspected manifold and reopen the valve connecting the fuel tank and the manifolds. If the leak has been eliminated, the system will quickly reach equilibrium. If the leak has not been isolated, the system will not reach a pressure equilibrium (at least, not before all of the fuel has drained out through the leak).

If FORTE begins with a correct system model along with the system behavior caused by a leak in the manifold, FORTE revises the model by deleting the constraint `minus(D_Amt_Fuel, D_Amt_Man)`. The variable `D_Amt_Fuel` is the amount of fuel leaving the fuel tank and flowing into the manifold. Variable `D_Amt_Man` is the net change in the amount of fuel in the manifold. Normally, the amount of fuel flowing out of the fuel tank should be the same, except for sign, as the net amount of fuel being added to the manifold. Since FORTE deletes this constraint, there must be another influence on the amount of fuel in the manifold, namely, a leak.

4 Conclusion

We have developed a number of systems for automatically refining imperfect knowledge bases by integrating various machine-learning methods. These systems have been successfully tested on a variety of real-world problems, including qualitative modelling of a complex subsystem of the Space Shuttle. We believe our results and those of other researchers in the area demonstrate the promise of automated knowledge base refinement. Hopefully, these methods will continue to be refined and successfully employed to speed the development of knowledge-based systems in additional application areas.

Acknowledgements

Thanks to Brad Richards, Dirk Ourston, Jeff Mahoney, and Paul Baffes as major contributors to the work reported in this paper. This research was supported by the NASA Ames Research Center

under grant NCC 2-629, the National Science Foundation under grant IRI-9102926 and the Texas Advanced Research Program under grant 003658114.

References

Baffes, P., and Mooney, R. (1993). Symbolic revision of theories with M-of-N rules. In *Proceedings of the Thirteenth International Joint Conference on Artificial intelligence*. Chambery, France.

Ehrenfeucht, A., Haussler, D., Kearns, D., and Valiant, L. (1989). A general lower bound on the number of examples needed for learning. *Information and Computation*, 82:267–284.

Ginsberg, A., Weiss, S. M., and Politakis, P. (1988). Automatic knowledge based refinement for classification systems. *Artificial Intelligence*, 35:197–226.

Kay, H. (1992). A qualitative model of the space shuttle reaction control system. Technical Report AI92-188, Artificial Intelligence Laboratory, University of Texas, Austin, TX.

Kuipers, B. J. (1986). Qualitative simulation. *Artificial Intelligence*, 29:289–338.

Ma, Y., and Wilkins, D. C. (1991). Improving the performance of inconsistent knowledge bases via combined optimization method. In *Proceedings of the Eighth International Workshop on Machine Learning*, 23–27. Evanston, IL.

Mahoney, J. J., and Mooney, R. J. (1993). Combining neural and symbolic learning to revise probabilistic rule bases. In Hanson, S., Cowan, J., and Giles, C., editors, *Advances in Neural Information Processing Systems, Vol. 5*, 107–114. San Mateo, CA: Morgan Kaufman.

Mahoney, J. J., and Mooney, R. J. (in press). Combining connectionist and symbolic learning to refine certainty-factor rule-bases. *Connection Science*.

Michalski, R. S., and Chilausky, S. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *Journal of Policy Analysis and Information Systems*, 4(2):126–161.

Mooney, R. J. (1993). Induction over the unexplained: Using overly general domain theories to aid concept learning. *Machine Learning*, 10:79–110.

Muggleton, S., and Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. In *Proceedings of the Fifth International Conference on Machine Learning*, 339–352. Ann Arbor, MI.

Muggleton, S. H., editor (1992). *Inductive Logic Programming*. New York, NY: Academic Press.

Ourston, D. (1991). *Using Explanation-Based and Empirical Methods in Theory Revision*. PhD thesis, University of Texas, Austin, TX. Also appears as Artificial Intelligence Laboratory Technical Report AI 91-164.

Ourston, D., and Mooney, R. (1990). Changing the rules: a comprehensive approach to theory refinement. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 815–820. Detroit, MI.

Ourston, D., and Mooney, R. J. (in press). Theory refinement combining analytical and empirical methods. *Artificial Intelligence.*

Porter, B., Bareiss, R., and Holte, R. (1990). Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45:229–263.

Quinlan, J. (1990). Learning logical definitions from relations. *Machine Learning*, 5(3):239–266.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Richards, B., and Mooney, R. (1991). First-order theory revision. In *Proceedings of the Eighth International Workshop on Machine Learning*, 447–451. Evanston, IL.

Richards, B., and Mooney, R. J. (1992). Learning relations by pathfinding. In *Proceedings of the Tenth National Conference on Artificial Intelligence.* San Jose, CA.

Richards, B. L. (1992). *An Operator-Based Approach To First-Order Theory Revision.* PhD thesis, University of Texas, Austin, TX. Also appears as Artificial Intelligence Laboratory Technical Report AI 92-181.

Rumelhart, D. E., Hinton, G. E., and Williams, J. R. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., and McClelland, J. L., editors, *Parallel Distributed Processing, Vol. I*, 318–362. Cambridge, MA: MIT Press.

Towell, G. G., Shavlik, J. W., and Noordewier, M. O. (1990). Refinement of approximate domain theories by knowledge-based artificial neural networks. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 861–866. Boston, MA.

Wisniewski, E. (1989). Learning from examples: The effect of different conceptual roles. In *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, 980–986. Ann Arbor, MI.

N94-34057

Improving the Explanation Capabilities of Advisory Systems[1]

Bruce Porter
Art Souther

Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

Abstract

A major limitation of current advisory systems (e.g., intelligent tutoring systems and expert systems) is their restricted ability to give explanations. The goal of our research is to develop and evaluate a *flexible* explanation facility, one that can dynamically generate responses to questions not anticipated by the system's designers and that can tailor these responses to individual users. To achieve this flexibility, we are developing a large knowledge base, a viewpoint construction facility, and a modeling facility.

In the long term we plan to build and evaluate advisory systems with flexible explanation facilities for scientists in numerous domains. In the short term, we are focusing on a single complex domain in biological science, and we are working toward two important milestones: 1) building and evaluating an advisory system with a flexible explanation facility for freshman-level students studying biology, and 2) developing general methods and tools for building similar explanation facilities in other domains.

[1]Support for this research was provided by the Air Force Office of Scientific Research (contract number F49620-93-1-0239) and the National Science Foundation (grant number IRI-9120310)

1 Research Objectives

The goal of our research is to develop and evaluate a *flexible* explanation facility that can dynamically generate responses to questions not anticipated by the system's designers and that can tailor these responses to individual users. Previous advisory systems have lacked these capabilities for a variety of reasons. In this section we will describe the problems of current advisory systems, the solutions to these problems that we propose, and our research activities for achieving those solutions.

Problems. The explanation facilities of current advisory systems are inflexible for two reasons:

- Inadequate domain knowledge: At least two factors limit the adequacy of the knowledge base as a source of "raw materials" for flexibly generating explanations: small size and task specificity. Although small size is an obvious limitation, few research projects have built a large-scale knowledge base as their "starting point" for research on explanation. Furthermore, because the knowledge for most advisory systems supports only a single task, most research on explanation has overlooked issues outside the task requirements, such as answering a range of questions, explaining terminology, and customizing explanations for specific users [22]. (For notable exceptions see work by Moore and Swartout [33, 24].)

- Inability to reorganize knowledge: Little work has been done to develop methods to select coherent packets of knowledge from a knowledge base, and even less on the reorganization of portions of the knowledge base to improve specific explanations. These issues have been avoided by "hardwiring" knowledge structures that are suitable for the limited explanations required by a particular advisory system. (For notable exceptions see work by McKeown [21] and Suthers [32].)

Solutions. We are developing a five-part solution to the problems of current advisory systems. Our solution comprises: (1) constructing a knowledge base which is large-scale and contains very fine-grained representations, (2) selecting and organizing knowledge with viewpoints and models, (3) generating new viewpoints on demand, (4) constructing and simulating models and using them to explain the behavior of mechanisms, and (5) generating explanations which relate new information to what the user already knows. We discuss each of these in turn.

First, we have built an extensive knowledge base for one area of biology — college-level anatomy and physiology of plants [26]. Although it is under constant development, it is already one of the largest knowledge bases in existence. (Our knowledge base currently contains about 3,000 frames and over 28,000 facts.) Unlike knowledge bases built with instructional frames [14] or hypertext [10], our knowledge base consists of "atomic facts" that our explanation facility can combine in different ways to produce different explanations.

Second, we are developing methods for selecting information from the knowledge base and organizing it into a coherent bundle appropriate to the situation at hand. One organizing structure is that of *viewpoints*, which provide coherent descriptions of objects or processes. For instance, the viewpoint "photosynthesis as a production process" selects and organizes facts to explain how photosynthesis produces glucose from carbon dioxide and water. Another organizing structure is that of *models*, which are built from viewpoints and support computer simulation. For example, an energy flow model of the plant includes the viewpoints "photosynthesis as an energy transduction process" and "respiration as an energy transfer process," and it allows an advisory system to predict and explain the effects of changes in light wavelength on a plant's photosynthetic or respiratory rate under a variety of specific circumstances.

Third, we are developing methods to automatically generate *new* viewpoints. This ability is important because, as system designers, we cannot anticipate all the viewpoints necessary for effective explanations. For example, Table 1 lists several viewpoints on photosynthesis and the situations in which they might arise. Our question answering facility will be able to construct these viewpoints by selecting and reorganizing the individual facts comprising existing viewpoints in the knowledge base (see [1]).

Fourth, we are developing methods for automatically constructing and simulating models and interpreting the consequences of simulations. These methods use existing methods of qualitative reasoning, but add two new capabilities: constructing models from large knowledge bases and generating explanations from these models. This will allow our explanation facility to answer "what-if" questions that were unanticipated when the knowledge base was built (see [28]).

Finally, we are developing methods to automatically generate *integrative explanations*, which explicitly relate new information to what the user already knows. This is important to advisory systems because the coherence of an explanation depends upon the particular situation. Our system will record the discourse with each user and will explain new topics

Viewpoint on Photosynthesis	*Contextual Situation*
as a destructive process	To explain the effects of the first oxygen producing plants on other organisms during evolution.
as an essential process in ecosystem energy flow	To explain how almost all living things depend on photosynthesis for deriving energy from an abiotic source.
as a magnesium-utilizing process	To explain the effects of magnesium deficiency on the plant.
as an enabling process	To explain how photosynthesis is important for any processes which use glucose or oxygen.
as a constructive process	To explain how photosynthesis is vitally important to plant growth and reproduction.

Table 1: A few of the viewpoints on photosynthesis and the teaching situations in which they might be appropriate.

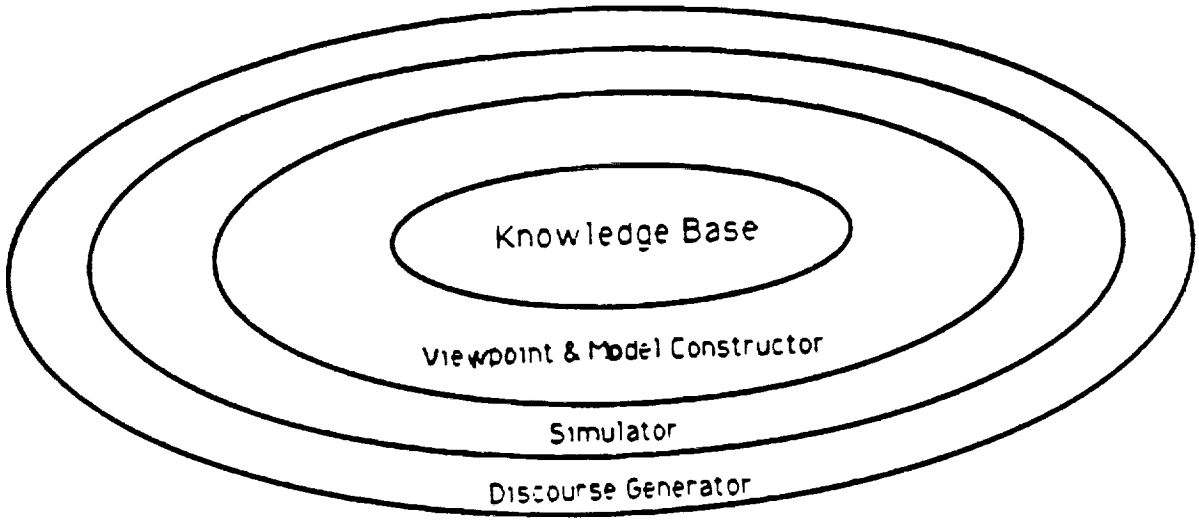


Figure 1: The layered design of our proposed advisory system. Each layer of software can access and use any layers within it.

in ways that relate to that user's knowledge and interests (see [18]).

2 The Design of Our Advisory System

An advisory system that simply provides facts to a user fails to take advantage of established techniques for effective communication. These techniques include treating the user as an active learner, grounding new information within a relevant context, and conveying information in appropriate ways through an interface which is intuitively easy to use. If the advisory system is to be used in a learning situation, it also needs to motivate the user with an appealing environment.

To provide these capabilities, our advisory system is designed with layers of software between the knowledge base and the user, each providing an essential capability for a flexible, reactive advisory system (see Figure 1). The outermost layer is the *discourse generator*, which interacts with the user by presenting focused information and encouraging the user to ask questions and to explore additional issues germane to the topic. To generate the relevant knowledge within an appropriate context and provide alternate modes of presentation, the discourse generator uses information from the inner layers. The *modeler and simulator*

predict and explain the behavior of biological systems by using computational models to answer "what if" and "why" questions; they permit the user to directly investigate the predictions of a model by manipulating its parameters. The *viewpoint constructor* selects and organizes domain information into coherent explanations. Many of these viewpoints may be directly encoded in the *knowledge base.* Others will be constructed by reorganizing the facts comprising existing structures.

This section describes the capabilities of each layer of software, and our current prototypes, beginning with the knowledge base.

2.1 A Knowledge Base for Biology

At the core of any advisory system is a knowledge base. It contains both the information to be communicated to the user and the information required for effective comunication, such as the background knowledge required to understand particular concepts.

For many domains, building a knowledge base is difficult and time consuming. To avoid this difficulty, most system designers have built advisory systems in subject areas for which a small knowledge base will suffice [35, 34, 4, 7, 6, 29, 16, 27, 25]. These subjects fall into two categories. The first is task-specific subjects that focus on a single application of knowledge. For example, the Guidon system [9] teaches diagnosis of infectious blood diseases. Teaching other tasks, such as how to determine a patient's prognosis, would require substantial changes to the system because Guidon is specialized for its single task. The second category of subjects is formally characterizable subjects that require only a small set of logical rules or axioms. For example, the GEOMETRY system [2] requires only a few rules of introductory geometry. However, the fundamental knowledge in a field like biology is neither committed to performing a single task nor formally characterizable with a small set of axioms. We believe that we con overcome the inherent difficulty in building a large knowledge base for two reasons: 1) we have developed sophisticated software that assists us in viewing and editing large, fine-grained knowledge bases; 2) we have used this software to build a large knowledge base, and applied our prototype systems for explanation generation to it.

2.2 The Viewpoint Constructor

A knowledge base for basic science must represent multiple *viewpoints* of each concept. For example, encoded in the Biology Knowledge Base are many different viewpoints of photo-

synthesis. Two of these, which we mentioned earlier, are "photosynthesis as a production process" and "photosynthesis as an energy transduction process." The knowledge base also contains more focused viewpoints that are appropriate in certain situations, such as "photosynthesis as a *glucose* production process" and "photosynthesis as an *oxygen* production process."

Figure 2 suggests why viewpoints are useful and even essential. The figure shows just part of the knowledge about photosynthesis that is encoded in our Biology Knowledge Base. Taken altogether, the totality of knowledge about photosynthesis is incoherent — there are so many facts about photosynthesis that some focus is necessary. Viewpoints provide this focus. The figure shows the two viewpoints of "photosynthesis as production" and "photosynthesis as energy transduction," highlighted with solid and dashed bold lines, respectively. Each collects and organizes facts about the basic process of photosynthesis that are relevant to that particular point of view and omits the large number of other facts that are irrelevant from that point of view. For example, "photosynthesis as production" focuses on the compounds, oxygen and glucose, that are produced by photosynthesis and on the compounds, carbon dioxide and water, that are its raw materials, and omits intermediate compounds, such as ATP that participate in photosynthesis but are, overall, neither produced nor consumed. This viewpoint also omits much other information about photosynthesis that is irrelevant to viewing photosynthesis as production.

A viewpoint, then, is a collection of facts about a particular concept that are all relevant within a particular context. The focus that viewpoints provide is critical because an arbitrary collection of facts is usually incoherent, even when the facts all pertain to the same topic. For example, describing photosynthesis as "a process that converts light energy into glucose and oxygen" is not patently incorrect but is confused or incoherent in that it intermixes facts from the viewpoints of energy flow and material flow. It is better to say that photosynthesis converts light energy into chemical bond energy (the energy transduction viewpoint), or that it converts carbon dioxide and water into glucose and oxygen (the production viewpoint). The *viewpoint constructor* is the part of our system that processes requests for viewpoints and produces the appropriate collection of facts selected from all facts in the knowledge base.

Many researchers acknowledge that viewpoints are a useful way of organizing knowledge. However, most methods for retrieving viewpoints from a knowledge base assume that each viewpoint is explicitly encoded [33, 23, 20]. Unfortunately, the difficulty of explicitly encoding viewpoints increases combinatorially with the number of concepts in the knowledge base. In

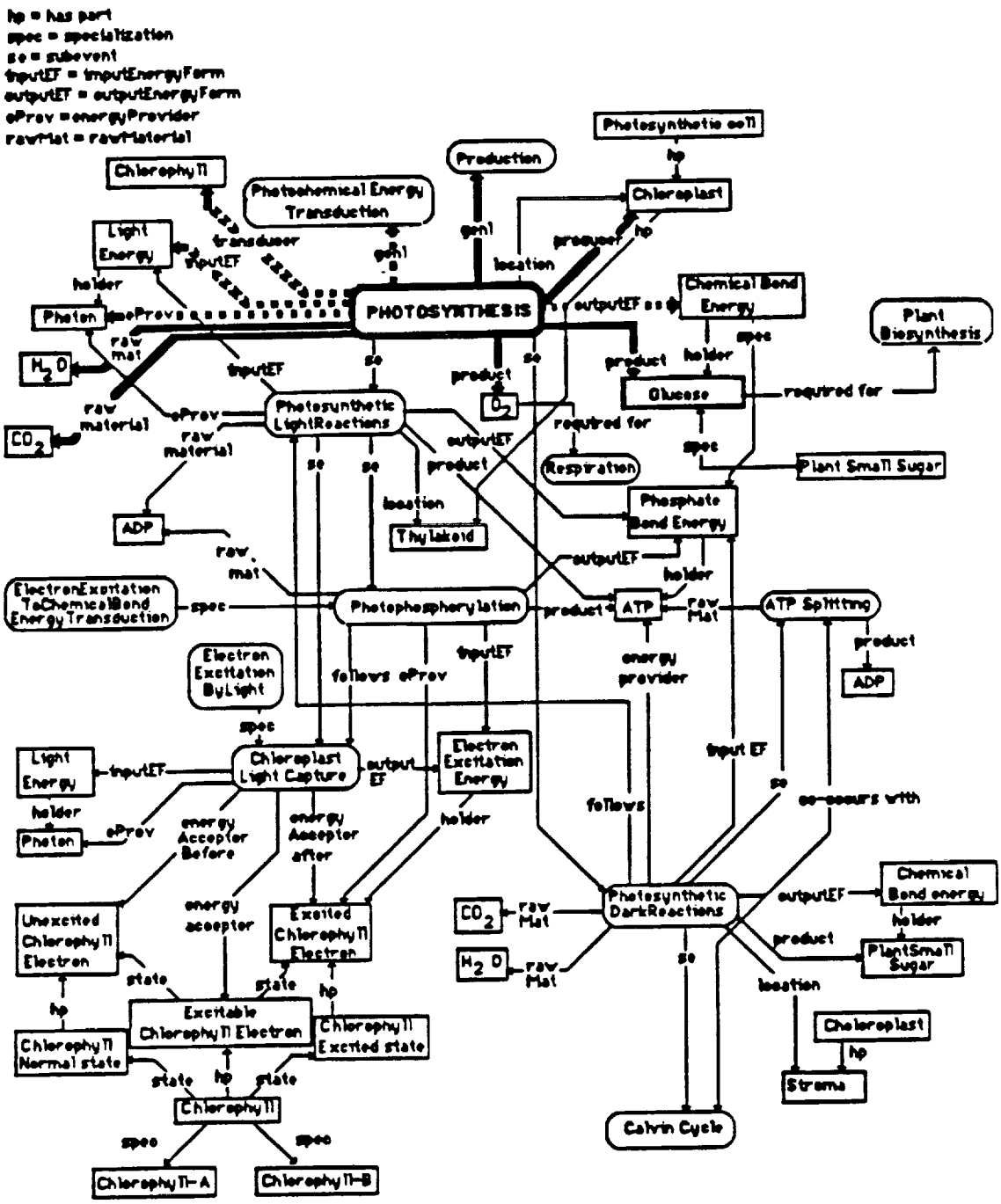


Figure 2: A small portion of the knowledge about photosynthesis represented in the Biology Knowledge Base. These labelled graphs, or "semantic networks", are widely used in artificial intelligence. Each fact is a relation (depicted as a labeled arc or line) between two concepts (depicted as labeled boxes). Solid bold lines represent information that is part of the viewpoint "photosynthesis as production", while the dotted bold lines represent the viewpoint of "photosynthesis as energy transduction".

addition, relying on pre-encoded viewpoints is inflexible because new viewpoints cannot be created as needed.

Our solution to this problem is to enable the advisory system to dynamically generate viewpoints when they are needed. We have experimented with methods for doing this using abstract specifications for points of view, called *view types*. For example, the *structural* view type specifies methods for constructing viewpoints concerning an object's parts and their interconnections, such as the viewpoint "endosperm as part of a seed." Similarly, the *functional* view type specifies methods for constructing viewpoints concerning the role of an object in a process, such as "chloroplast as the producer in photosynthesis."

View types can also be combined. The *structural-functional* view type specifies how the individual parts of an object participate in the subevents of some process. For example, a structural-functional description of angiosperm sexual reproduction would discuss how each part of the flower (sepals, petals, stamen, and carpels) participates in some event of the reproductive process (e.g., pollinator attraction, pollen formation, and pollination).

We believe that a relatively small number of such view types is sufficient to characterize and produce many viewpoints within the natural sciences. Support for this conjecture is preliminary but encouraging. First, we found that our view types and their combinations are sufficient to characterize over fifty definitions chosen at random from the glossary of a biology textbook. Second, we have successfully used view types in a prototype system for generating viewpoints [1, 30]. These viewpoints constitute answers to a wide range of definational questions (e.g., "What is C3-photosynthesis?") and comparative questions (e.g., "What is the difference between mitosis and meiosis?").

2.3 The Modeler and Simulator

Our advisory system will use computational models to predict and explain the behavior of complex biological systems. This capability is very important because it can tie together otherwise disparate and uninteresting facts into an explanation of how something works.

Most computational models in biology are *quantitative* models, which interrelate a system's parameters using differential equations. Although these models are precise, they can also be intractable, especially if some of the equations are nonlinear. Moreover, because quantitative models require complete numeric data, model builders must assume precise values for parameters for which little precise data may be known. Finally, the quantitative details often obscure the more important qualitative principles.

During the past ten years, research on *qualitative* models has addressed these problems [15, 13, 11]. Instead of using exact relationships and values, qualitative models employ qualitative relationships, such as "water potential increases with turgor," and qualitative values, such as "cell turgor is positive and decreasing." Approximations like these are frequently sufficient to express essential information about a system when complete knowledge is unavailable or unneccessary. They also enable a qualitative simulator to characterize the behavior of a system, much as a human reasoner could, without knowing or needing exact relationships or values. For example, a qualitative simulator with a model of a plant's water flow could predict that "excessive transpiration from a plant caused by increasing temperatures will be countered by closing of the stomata" without knowing the original concentration of water in the plant or the exact rate of transpiration. Qualitative models have been used in advisory systems for steam-plant operation [31], weather prediction [5], circuit diagnosis [3, 35], and many other domains.

We are extending the research on qualitative reasoning in two ways. First, while previous research assumes that a model is given *a priori*, we are developing methods for constructing models as needed. In order to support a wide range of questions, our knowledge base must provide a vast array of viewpoints and levels of detail. However, overly detailed models, while perhaps capable of answering many questions, can be inefficient or even intractable, and excess detail would make their predictions opaque. Our program uses each question to decide which perspectives and abstractions are needed, constructs a model from these pieces, and simulates this model to answer the question (see [28]). Such a model not only answers the question, but also highlights the knowledge supporting the answer and provides transparent, explainable answers.

Second, we are developing methods to generate in-depth explanations of qualitative reasoning. A major shortcoming of current simulators (both qualitative and quantitative) is that they generate extensive details about a model's behaviors but little overview or explanation. Our system will provide concise and focused textual answers to a range of questions about a model and its behaviors. For example, we expect to provide multilevel overviews of both a model and its behaviors which highlight their most important features and compare and contrast different behaviors (if there is more than one). We also expect to provide an explanation of the mechanisms by which a model causes its behaviors, grounded in familiar physical principles, and how a model would respond to changed circumstances (see [19]).

2.4 The Explanation Generator

Our overriding goal is to develop and evaluate a *flexible* explanation facility that can dynamically generate responses to questions not anticipated by the system's designers and that can tailor these responses to individual users. We are building an explanation generator that will achieve flexibility in three ways. First, it will produce *integrative explanations* that relate new information to the user's existing knowledge. In producing an integrative explanation, we can define three networks of relevant concepts and relations. The *target* network is the set of concepts and relations that a system seeks to communicate to the user. The *base* network is the set of concepts and relations that model what the user already understands and is relevant in some way to the target. The *linking* network is the set of concepts and relations that relate the target to the base. To produce an integrative explanation, our system will determine the relevant target, linking, and base networks, and it will organize the knowledge in the linking and target networks in a manner that facilitates their integration into the base network.

Opportunism is the second way that our explanation generator will achieve flexibility. The system will actively seek opportunities to include important information in the domain that is closely related to the topic being explained but is unknown to the user. For example, suppose the system were explaining embryo sac formation to a user, and noticed that two participants in this process, a megaspore and a megaspore mother cell, are both kinds of botanical cells. It can recognize this as an opportunity to discuss the difference between haploid and diploid cells, an important distinction in biology. Moreover, rather than interjecting this discussion in the middle of another topic, the system can relocate it to an appropriate place in its explanation.

Finally, our explanation generator will achieve *organizational flexibility.* Such flexibility is desirable for two reasons. First, a generator should be able to introduce prerequisite material and elaborations at appropriate positions in the explantion. Second, it should be able to place material that is familiar to the user earlier in the explanation and material that is new to the user later. To achieve organizational flexibility, the generator takes a *delayed-commitment* approach: it delays organizational commitments as long as possible. Initially, the propositions of the explanation are organized very loosely. As the explanation develops, the generator adds new propositions and gradually arranges them in an order that is most suitable for the user.

We are aided in our efforts to construct an explanation generator by previous research

results on user modeling and natural language generation. An *overlay* model [8] represents what the user knows as a subset of the concepts in the knowledge base. The explanation generator initializes the user model with basic concepts covered in previous courses and lessons, and updates the model based upon explanations that it generates and questions the user asks. Also, we are using the FUF system [12] for converting explanation structures into English. FUF, which has been in development at Columbia for the past seven years, employs one of the largest machine grammars ever constructed and provides wide linguistic coverage.

We have constructed a prototype system, which provides integrative explanations, opportunism, and organization flexibility [17, 18]. We have used this system to produce multi-paragraph explanations from portions of the Biology Knowledge Base. Because the system is not restricted to schemas, it generates different explanations for different users. The system's output was favorably evaluated by a domain expert, who found the explanations both accurate and clear.

3 Evaluating and Generalizing Our Results

Our long-term objective is to build advisory systems for complex domains that compete well with human advisors. Although we cannot meet this objective soon, we believe we can build and evaluate the core components of an advisory system that competes well with textbooks for an important portion of a course, and that meeting this short-term objective is a critical milestone for achieving our long-term objective.

We plan to evaluate our advisory system by using it to help teach an introductory biology course at the University of Texas at Austin. In addition to introductory material, the system will explain advanced material that has not been covered in the classroom or assigned readings.

The evaluation will be based on data from the following experiment. Users will be paid to spend extra time in the course studying the advanced material with the help of the advisory system. When the users are comfortable using the system, we will give them several assignments. Each assignment will require answers and explanations for a range of technical questions on both the introductory and advanced material. (These questions will be formulated by a biologist who is not affiliated with our project. Our research team will not know the questions beforehand.) To complete their assignments, the users will be randomly assigned to three groups. Users in the "traditional" group will be permitted to

use any standard (non-human) resources, such as textbooks and laboratory equipment. The "advisory" group will be allowed to use only the advisory system, and the "eclectic" group will be allowed to use both traditional sources and the system.

We will compare the performance of the three groups of users on correctness and completeness of answers and on efficiency of task completion. The users' answers and explanations will be judged by the teaching staff for the biology course, who will not be apprised of the users' identity or group. If a benefit for the advisory system is found, we will separately analyze user performance on the introductory material to see if a benefit exists even when the material has been covered in the classroom. Including the eclectic group will further allow us to ascertain whether there is a synergistic effect among the three sources of information — classroom, textbook, and advisory system. The users' proficiency in terms of the amount of time used to complete the assignment will be measured, controlling for the correctness of the users' responses. For each of the three groups, we will also measure the users' interest in the advanced materials taught. This assessment will be based on questions from standard course evaluations.

Based on the results of our evaluation, we will generalize our research results to help others build advisory systems in a range of domains. This will involve removing dependencies on the domain of biology that our experience will no doubt reveal and re-implementing those parts of our system that contributed most to its success, to improve its portability and ease of reuse.

4 Summary

The primary results of this research will be the following: (1) an explanation facility for college-level biology, (2) a critical evaluation of the explanation facility based upon its use in an introductory biology course at the University of Texas, and (3) general methods and tools for building similar explanation facilities in other domains.

During the last six years, we have built a very large knowledge base for one area of biology and we have developed prototype systems for each component of our proposed explanation facility. From this experience, we have learned how to structure large knowledge bases using viewpoints and models, and we have created a foundation on which to build a flexible explanation facility.

Our proposed explanation facility will dynamically generate responses to unanticipated

questions and tailor these responses to individual users. This flexibility will encourage a user to ask questions and request clarification or detail. In the future we expect this functionality to be the foundation for a wide range of computer-based advisory and research tools, such as intelligent databases, electronic libraries, and simulated laboratories.

References

[1] L. Acker and B. Porter. Access methods for large knowledge bases. *Artificial Intelligence*, (submitted), 1992.

[2] J. Anderson, C. Boyle, and G. Yost. The geometry tutor. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1–7, 1985.

[3] J. Brown, R. Burton, and A. Bell. Sophie: A step towards a reactive learning environment. *International Journal of Man-Machine Studies*, 7:675–696, 1975.

[4] J. Brown, R. Burton, and J. de Kleer. Pedagogical, natural language, and knowledge engineering techniques in Sophie I, II, and III. In D. Sleeman and J. Brown, editors, *Intelligent Tutoring Systems.* Cambridge, MA: Academic Press, 1982.

[5] J. Brown, R. Burton, and F. Zdydel. A model-driven question-answering system for mixed-initiative computer-assisted instruction. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:248–257, 1973.

[6] R. Burton. Diagnosing bugs in a simple procedural skill. In *Intelligent Tutoring Systems.* London: Academic Press, 1982.

[7] R. Burton and J. Brown. An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, 11:5–24, 1979.

[8] B. Carr and I. Goldstein. Overlays: A theory of modelling for computer aided instruction. Technical Report AI Lab Memo 406 (Logo Memo 40), Massachusetts Institute of Technology, 1977.

[9] W. Clancey and R. Letsinger. Neomycin: Reconfiguring a rule-based expert system for application to teaching. In *Seventh International Joint Conference on Artificial Intelligence*, pages 829–836, 1981.

[10] J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(9):17–41, 1987.

[11] J. DeKleer and J. Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24:7–83, 1984.

[12] M. Elhadad. *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation.* PhD thesis, Department of Computer Science, Columbia University, 1992.

[13] K. Forbus. Qualitative process theory. *Artificial Intelligence*, 24:85–168, 1984.

[14] G. Kearsley. Intelligent computer-aided instruction. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 154–159. John Wiley and Sons, New York, 1987.

[15] B. Kuipers. Qualitative reasoning: Modeling and simulation with incomplete knowledge. *Automatica*, 25(4):571–585, 1989.

[16] P. Langley, J. Wogulis, and S. Ohlsonn. Rules and principles in cognitive diagnosis. In N. Frederiksen, editor, *Diagnostic Monitoring of Skill and Knowledge Acquisition.* Hillsdale, New Jersey: Lawrence Earlbaum Associates, 1987.

[17] J. Lester and B. Porter. Generating integrative explanations. In *Proceedings of the AAAI Workshop on Explanation*, pages 80–89, 1990.

[18] J. Lester and B. Porter. A revision-based model of instructional multi-paragraph discourse production. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, August 1991.

[19] R. Mallory. Generating structured, causal explanations of qualitative simulations: A research proposal. Technical Report (forthcoming), Department of Computer Science, University of Texas at Austin, 1993.

[20] K. McCoy. The role of perspective in responding to property misconceptions. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 791–793, 1985.

[21] K. McKeown. Discourse strategies for generating natural language text. *Artificial Intelligence*, 27:1–42, 1985.

[22] K. McKeown and W. Swartout. Language generation and explanation. *Annual Review of Computing Science*, 2:401–409, 1987.

[23] K. McKeown, M. Wish, and K. Matthews. Tailoring explanations for the user. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 794–798. Palo Alto, California: Morgan Kaufmann, 1985.

[24] J. Moore and W. Swartout. A reactive approach to explanation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 1504–1510, 1989.

[25] B. Porter, R. Bareiss, and R. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence Journal*, 45(2):229–263, 1990.

[26] B. Porter, J. Lester, K. Murray, K. Pittman, A. Souther, L. Acker, and T. Jones. AI research in the context of a multifunctional knowledge base: The Botany Knowledge Base project. Technical Report AI88-88, University of Texas at Austin, 1988.

[27] B. Reiser, J. Anderson, and R. Farrell. Dynamic student modeling in an intelligent tutor designed for LISP programming. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 8–14, 1985.

[28] J. Rickel and B. Porter. Using interaction paths for compositional modeling. In *Proceedings of the Sixth International Workshop on Qualitative Reasoning about Physical Systems*, pages 82–95, 1992.

[29] D. Sleeman. Inferring student models for intelligent computer-aided instruction. In *Machine Learning: An Artificial Intelligence Approach*, pages 483–510. Palo Alto, California: Morgan Kaufmann, 1983.

[30] A. Souther, L. Acker, J. Lester, and B. Porter. Using view types to generate explanations in intelligent tutoring systems. In *Proceedings of the Eleventh Cognitive Science Conference*, pages 123–130, 1989.

[31] A. Stevens and B. Roberts. Quantitative and qualitative simulation in computer-based training. *Journal of Computer-Based Instruction*, 10(1):16–19, 1983.

[32] D. Suthers. Providing multiple views of reasoning for explanation. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, pages 435–442, 1988.

[33] W. Swartout. Xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21(3):285–325, 1983.

[34] K. vanLehn and J. Brown. Planning nets: a representation for formalizing analogies and semantic models of procedural skills. In R. Snow, P. Frederico, and W. Montague, editors, *Aptitude, learning, and instruction: Cognitive process analyses*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

[35] B. White and J. Frederiksen. Qualitative models and intelligent learning environments. In R. Lawler and M. Yazdani, editors, *AI and Education*. New York: Ablex Publishing Co., 1987.

N94-34058

Recursive Heuristic Classification

David C. Wilkins
University of Illinois

The author will describe a new problem-solving approach called recursive heuristic classification, whereby a subproblem of heuristic classification is itself formulated and solved by heuristic classification. This allows the construction of more knowledge-intensive classification programs in a way that yields a clean organization. Further, standard knowledge acquisition and learning techniques for heuristic classification can be used to create, refine, and maintain the knowledge base associated with the recursively called classification expert system. The method of recursive heuristic classification was used in the Minerva blackboard shell for heuristic classification. Minerva recursively calls itself every problem-solving cycle to solve the important blackboard scheduler task, which involves assigning a desirability rating to alternative problem-solving actions. Knowing these ratings is critical to the use of an expert system as a component of a critiquing or apprenticeship tutoring system. One innovation of this research is a method called dynamic heuristic classification, which allows selection among dynamically generated classification categories instead of requiring them to be prenumerated.

Session A4: ARTIFICIAL INTELLIGENCE IV

Session Chair: Dr. Abe Waksman

N94- 34059

Agent Oriented Programming

Yoav Shoham
Stanford University
Stanford, CA

The goal of our research is a methodology for creating robust software in distributed and dynamic environments. The approach taken is to endow software objects with explicit information about one another, to have them interact through a commitment mechanism, and to equip them with a speech-acty communication language. System-level applications include software interoperation and compositionality. A government application of specific interest is an infrastructure for coordination among multiple planners. Daily activity applications include personal software assistants, such as programmable email, scheduling, and new group agents. Research topics include definition of mental state of agents, design of agent languages as well as interpreters for those languages, and mechanisms for coordination within agent societies such as artificial social laws and conventions.

N94-34060

PI IN THE SKY: THE ASTRONAUT SCIENCE ADVISOR ON SLS-2

Lyman R. Hazelton
Center for Space Research
Massachusetts Institute of Technology
Cambridge, MA 02139-4307

Nicolas Groleau
Richard J. Frainier
Michael M. Compton
RECOM Technologies, Inc.
Artificial Intelligence Research Branch
NASA Ames Research Center
Mail Stop 269-2
Moffett Field, CA 94035-1000

Silvano P. Colombano
Artificial Intelligence Research Branch
NASA Ames Research Center
Mail Stop 269-2
Moffett Field, CA 94035-1000

Peter Szolovits
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139-4307

Abstract

The Astronaut Science Advisor (ASA, also known as Principal-Investigator-in-a-Box) is an advanced engineering effort to apply expert systems technology to experiment monitoring and control. Its goal is to increase the scientific value of information returned from experiments on manned space missions. The first in-space test of the system will be in conjunction with Professor Larry Young's (MIT) vestibulo-ocular "Rotating Dome" experiment on the Spacelab Life Sciences 2 mission (STS-58) in the Fall of 1993. In a cost-saving effort, off-the-shelf equipment was employed wherever possible. Several modifications were necessary in order to make the system flight-worthy. The software consists of three interlocking modules. A real-time data acquisition system digitizes and stores all experiment data and then characterizes the signals in symbolic form; a rule-based expert system uses the symbolic signal characteristics to make decisions concerning the experiment; and a highly graphic user interface requiring a minimum of user intervention presents information to the astronaut operator. Much has been learned about the design of software and user interfaces for interactive computing in space. In addition, we gained a great deal of knowledge about building relatively inexpensive hardware and software for use in space. New technologies are being assessed to make the system a much more powerful ally in future scientific research in space and on the ground.

PI IN THE SKY: THE ASTRONAUT SCIENCE ADVISOR ON SLS-2

Introduction

The Astronaut Science Advisor (originally called "Principal Investigator-in-a-Box", abbreviated [π]) project is an application of Expert Systems technology from the field of Artificial Intelligence to the conduct of space science experiements. It aim is to improve the quality and yield of experimental science on current shuttle missions and long duration missions of the type foreseen for the Space Station. It encapsulates in a computer program some of the experiment related knowledge and reasoning possessed by the Principal Investigator. The primary user of the system is the astronaut performing the experiment, but reference to the system by the Principal Investigator and possibly by the Mission Manager is also envisioned.

Scientific research is conducted to elucidate unknown quantities and processes in nature. The first step in doing research is the construction and recording of a hypothetical model (a theory) which might describe a process or define a quantity. An important feature of a good theory is that it should be *testable*. This means one should be able, based on one's theory, to suggest one or more experiments, the outcomes of which are clearly predicted in advance. The validity of the theory is then verified by the expected experimental outcome.

This rather simple description ignores the real complexity of doing modern scientific research. The systems under study today are almost always too complex to approach with a finished theory and some "make or break" experiment. Instead, scientists create a preliminary theory which can be tested "by parts" and "tuned". An experiment is carried out a few steps at a time, all the while noting whether the system is behaving in a way consistent with the predictions of the theoretical model. If the model seems correct, the experiment continues along the lines initially constructed from the theory. If there is disagreement between experimental observation and the theory predictions, the theory is *modified* by the scientist so it more closely predicts what has been observed. Such alteration of the system's model generally requires modification of the experimental procedure before continuing the investigation. The research process continues, iteratively, in this manner until the scientist is convinced no further information will be obtained with the current experiment. The resulting new theory is announced and, perhaps, new experiments are proposed based on it.

It is very difficult to do scientific research in space because most of the time the scientist(s) are not among the flight crew. Instead, a carefully chosen and highly trained "best fit" crew flies with the experiments while the scientists remain on the ground. When possible, real-time sent to the scientists while their experiments are active. However, as the size and complexity of the experimental environment increases, the availability of communication bandwidth for real-time ground data acquisition decreases. Furthermore, the scientist on the ground may not be able to communicate complex changes in an experiment protocol in time to have the crew implement them in the current experiment session. Finally, due to orbital geometry and the limited number of Tracking and Data Relay Satellites, there are periods of "loss of signal" during which no data are available on the ground. The recorded on orbit for later transmission, but generally does not become available until the night after the experiment was executed. Most of the time experiment protocols are performed in their original form because of these limitations. They are not executed in part and modified as is good scientific practice. If, as often happens, post flight analysis of the data indicates a requirement to change the theory describing a system, the only recourse available to the scientist is to propose another flight of the experiment in order to test the new model. This method of doing scientific research in space is both expensive and exasperatingly slow.

The Astronaut Science Advisor effort is a first step at circumventing these limitations and improving the scientific return from experiments done in space. The idea is to fly a computer system which has some of the expert factual knowledge and decision making ability of the scientist together with real time data acquisition for a large number of signals and a highly intuitive and informative human interface. It is effectively a limited alter ego of the Principal Investigator. This computer system contains a rudimentary representation of the theoretical model and a mechanism to make comparisons of observations (obtained via the data acquisition portion) with model predictions. It also contains a system to create and suggest alterations to the initial protocol if advantageous.

The version of the [π] being flown on SLS-2 is knowledgeable about Professor Larry Young's Rotating Dome Experiment. It will record all electronic data produced by the experiment and act as a "watch dog" to ensure the experimental apparatus is operating correctly and the not corrupted by malfunctioning equipment. It will analyze

specific portions of the data with respect to a theoretical model and, on demand from the astronaut operator, suggest alternative protocols designed to maximize the utility of the information being produced. Finally, the computer is aware of the time allocated for the protocol and indicates how closely the experiment is keeping to its schedule. Should the experiment run significantly late, the astronaut can be provided with a revised protocol which is designed to gain the most important information possible in the remaining time.

The applicability of the technology being developed for the Astronaut Science Advisor is not limited to manned research in space. It can be applied to medical diagnosis and research (see [GRO92]). A remote data collection and monitoring facility such as might be used for oil wells which are not visited for long periods could benefit from this technology. We believe this kind of system can greatly increase the productivity of unmanned planetary explorer missions by increasing the ability of the system to quickly respond to environmental changes and by decreasing the telemetry load. The development of this technology is still in its infancy. It is clear other applications will appear as it matures.

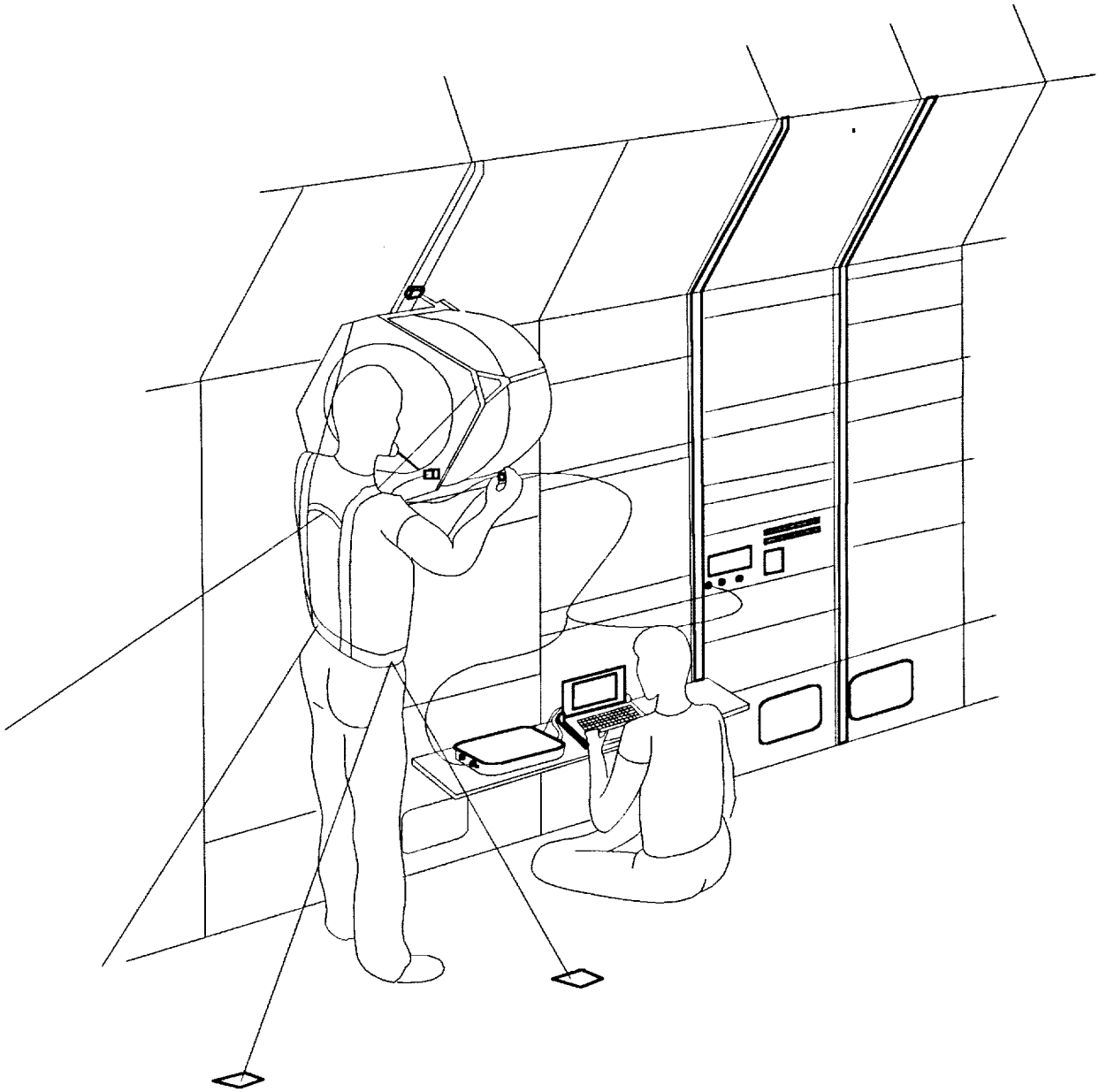


Figure 1

PI IN THE SKY: THE ASTRONAUT SCIENCE ADVISOR ON SLS-2

Background: The Rotating Dome Experiment

The Rotating Dome Experiment (see Figure 1) is designed to study the human balance system. The sensory inputs for balance are produced by the visual, vestibular, and *proprioceptive* systems. The proprioceptive sense indicates the relative positions of, and forces acting upon the various parts of the body. A model of the human balance system exists (see [YOU84]). However, the parameters of the model, particularly the weights of the different sensory inputs in the overall estimation of position, are not well defined. Even the general structure of the model is difficult to determine. The difficulty arises because it is practically impossible to decouple the different inputs to study the effect of one at a time.[1] Performing the experiment in the micro-gravity environment removes all but the visual and proprioceptive inputs, theoretically allowing a better determination of model parameters and structure.

Specifically, the experiment exposes a human subject to a rotating visual field in the roll axis. The roll axis passes approximately from the back of the subject's head through the tip of the subject's nose. To a varying degree, after a short time subjects begin to feel as if they are rotating while the visual field is perceived to slow or even stop. This perception of motion in the absence of real, physical motion is called *vection*. There are measurable subjective and physiological responses to roll vection, including involuntary twisting of the eyeball in its socket (ocular torsion), tilting of the head, and a general sway of the entire torso.

On orbit, the experiment will be carried out under three conditions: free-floating, the subject biting on the biteboard; tethered, the subject floating completely but held within a small volume of space by a set of loose tethers; and bungeed, weight upon the subject's feet simulated by attaching a set of bungee cords from the floor to a torso harness.

There is an additional reason for interest in performing the Rotating Dome experiment while on orbit. Many astronauts feel ill, and some become severely sick, while in the micro-gravity environment. This *space motion sickness* is expensive, dangerous, and poorly understood. It is thought a major cause of motion sickness is conflict among the position sensory inputs involved with balance. Indeed, some subjects of the Rotating Dome experiment experience motion sickness even while on the ground. A better model of the balance system might shed some light on ways to combat or eliminate space motion sickness.

Dome Data

Data collected during the experiment consist of:

1. Dome tachometer. The dome provides a coded square wave tachometer signal which allows determination of both its speed and direction of rotation.
2. Subjective estimation of vection. The subject is provided with a small one-dimensional, spring-centered joystick with which to indicate his or her relative level of vection. Full deflection indicates the subject feels the visual field (the dome) is not moving at all, while the subject is rotating. No deflection indicates the subject feels stationary while the visual field is moving.
3. Biteboard torque. Individually molded biteboards are anchored to a fixed truss in the dome by a strain gauge bridge. The subject may secure him- or herself in the dome by biting on the biteboard. Any tendency to tilt the head will be translated into changes in the strain gauge output.
4. Electro-myograph signals. Two skin contact electrodes are adhesively attached to each side of the subject's neck over the thickest part of the sterno-clavicular mastoid muscles. The electrodes are connected to high gain physiological amplifiers. The system allows recording of motor neuron pulse activity associated with contractions of the muscles involved in head tilt.

---

[1] Some work in this area has been done with animals through selectively destroying the nerves and/or organs associated with one or more of the senses involved.

5. Ocular video. The center of the rotating dome has a hole through which a video camera may be focused to produce a close-up image of the subject's right eye. The subject wears a specially prepared soft contact lens with fiducial marks which allow measurement of ocular torsion. Putting a drop or two of distilled water into the eye, which makes the surface of the sclera sticky for a short time, prevents the normal "floating" motion of the lens.
6. Body sway video. A second video camera is located behind the subject to provide a record of body sway due to involuntary response to vection. Both cameras also provide time stamping to allow synchronization with the electronic data during analysis.

The Rotating Dome experiment has flown on three previous orbiting missions (SL-1, D-1, and SLS-1). It is controlled by an Experiment Control and Data System (ECDS) computer, a space rated Digital Equipment Corporation PDP-8. The ECDS has a very limited program which sets the rotation modes of the dome and turns on and off the dome motor at the appropriate times. It also converts the analog dome signals described above to digital form and presents the resulting data stream to a high rate multiplexor for real-time transmission to the ground. The also recorded for re-transmission in case initial transmission fails. The re-transmission process can be controlled from the ground and generally takes place during the astronauts' sleep periods. A small, battery powered, two channel oscilloscope is also available to the astronauts to see the analog signals at the ECDS inputs if necessary. It should be noted that the ECDS does *no* analysis of the experiment data, and the oscilloscope is not of the storage variety, so the astronauts can never see the overall results of even one complete trial using this equipment. They essentially have no feedback from the standard equipment about how well or poorly the experiment is being performed.

It cannot be overly emphasized that this experiment involves individual physiological responses. Very large variation in any population to identical stimuli requires each subject be viewed as a separate experiment. While comparisons between subjects may elicit interesting population-wide trends, the width of the distribution makes the validity of such conclusions highly suspect. Meaningful conclusions can only be made by comparison of observed differences of individual responses in- and outside of the effects of gravity. It is precisely this broad variation in individual response which makes the experiment difficult. While it may be interesting to pursue a repetition of part of the experiment to verify data from one subject under a given condition, it may be a waste of time to test any of the other subjects in the same manner. It is here scientific expertise becomes imperative.

[π] Hardware

The hardware architecture of [π] consists of two parts: a computer and an analog interface box.

"Off-the-shelf" equipment is employed in the system wherever possible. Constraints requiring modification or outright fabrication are, however, abundant. The entire system has to fit into half of one stowage drawer (approximately 33 x 29 x 13 cm). It has to be rapidly deployable and easily re-stowed in the zero-g environment. It cannot require more than 90 watts power. It must pass stringent safety, off-gassing, and conducted and radiated electromagnetic interference tests. Finally, any failure, due to hardware or software, must have no effect on the Rotating Dome experiment.

The computer is a flight-modified version of an Apple Macintosh PowerBook 170 laptop. Its memory is augmented to the maximum allowed, eight mega-bytes. The choice of Apple's PowerBook 170 was predicated on four years of software development on Apple computers together with the unit's small size, low mass, and low power requirements. We were fortunate because other experimenters were interested in using this computer in the manned space program. A small consortium was formed to share the cost to determine modifications required for limited flight qualification and have them implemented. The modifications to the Macs (one for flight, one flight backup, and one for testing by the development team) were done by a special laboratory at Johnson Space Center.

An on-going concern is the thermal energy produced by the machine. The laptop's 68030 microprocessor is a CMOS device. The more active the device (i.e., the less time the processor is idling), the more thermal energy it produces. Execution of [π] comes very close to utilizing every available cycle during both data acquisition and

inter-trial analysis. Measurement of the temperature rise of the processor in the laboratory indicates the system will be functioning on orbit very close to the published maximum operating temperature. The problem of thermal balance is caused by the fact there is no convective cooling in the microgravity environment. We believe thermal overloading is a possible hardware failure mode for the system. Its use will be limited to a few hours on each of three days during the mission, so we believe the probability of failure is remote.

The Analog Interface Box (AIB) contains a power supply for the Macintosh, an eight channel high impedance analog to digital converter (A/D), a power supply for the A/D, and a Small Computer Systems Interface for communication with the computer (GW Instruments, Sommerville, Massachusetts). Power for the computer and A/D is drawn from Spacelab's standard 28 Volt DC bus via a tap in the rotating dome lighting circuit. The AIB's housing is a 7.8 pound machined solid aluminum box which acts as both electromagnetic shield and heat sink. New cabling was designed and fabricated to allow access to the analog data produced by the Dome.

[π] is considered a non-critical addition to the Rotating Dome experiment. This played a major role in reducing the cost and development time for the system. Standard (Class C) Spacelab hardware and software for experiment critical purposes is required to meet such rigorous fabrication and testing demands each piece must be individually produced. This increases the cost, even compared to modified off-the-shelf items, by at least an order of magnitude. [π] must interface with a critical experiment data path. However, by designing and fabricating just the interface to Class C standards and guaranteeing any failure on the [π] side of the interface will not cause interference to the host experiment, the rest of the computer system was accepted at Class D certification. While this still included a number of expensive hurdles which had to be jumped, the most severely demanding and expensive ones were eliminated.

[π] Software

NASA Life Sciences and Mission Management demanded [π] must not extend the time necessary to execute the Rotating Dome experiment. This single constraint drove many of the system design decisions. Operator inputs to the computer were kept to an absolute minimum and the system was designed to optimize the order of protocol steps to eliminate repeated operations wherever possible. If synchronization with the ECDS is lost or a dome malfunction of an unexpected nature occurs, the operator can enter a special oscilloscope mode. In this mode, [π] continuously displays data from all five channels without regard to dome rotation and without recording data. Should the [π] system itself be perceived to fail for any reason, the crew is instructed to power-down the computer and return it to its stowage drawer. This "Sword of Damocles" hanging over the experiment acted as a great incentive to quality assurance and verification of both hardware and software. In addition, astronaut comments and suggestions for changes and improvements were actively pursued and integrated into the system. The team spent a considerable portion of both time and monetary budgets on crew training and evaluation. In all, the software went through five releases before the final flight version was submitted.

The overall software architecture of [π] is best described as three major, independent but interacting modules.

First, a module written in the LabVIEW (National Instruments, Austin, Texas) language controls the A/D conversion and stores the resulting data in appropriate arrays. This module also does analysis of the numerical data to produce a small set of characteristic numbers or symbols describing the results of an experiment trial.

Second, a forward-chaining inference system written in CLIPS (NASA) uses the symbolic information provided by the first stage with a static rule base to infer decisions about the experiment. In particular, at the beginning of each experiment session the Rotating Dome system is subjected to a functional test sequence. Data from the functional test is used to determine the operational status of the dome and AIB hardware and to ascertain the "null" values for the various signals. The latter step is important because there are small, but important differences in the values of some of the components in the various versions of the dome hardware. We have no way of knowing ahead of time which instance of the dome hardware will actually be flown on the mission. Experiment-time determination of the "null" values for each of the signals allows the system to automatically compensate for these

differences as well as any changes which might occur due to the equipment being stored for several months in the Spacelab module prior to launch.

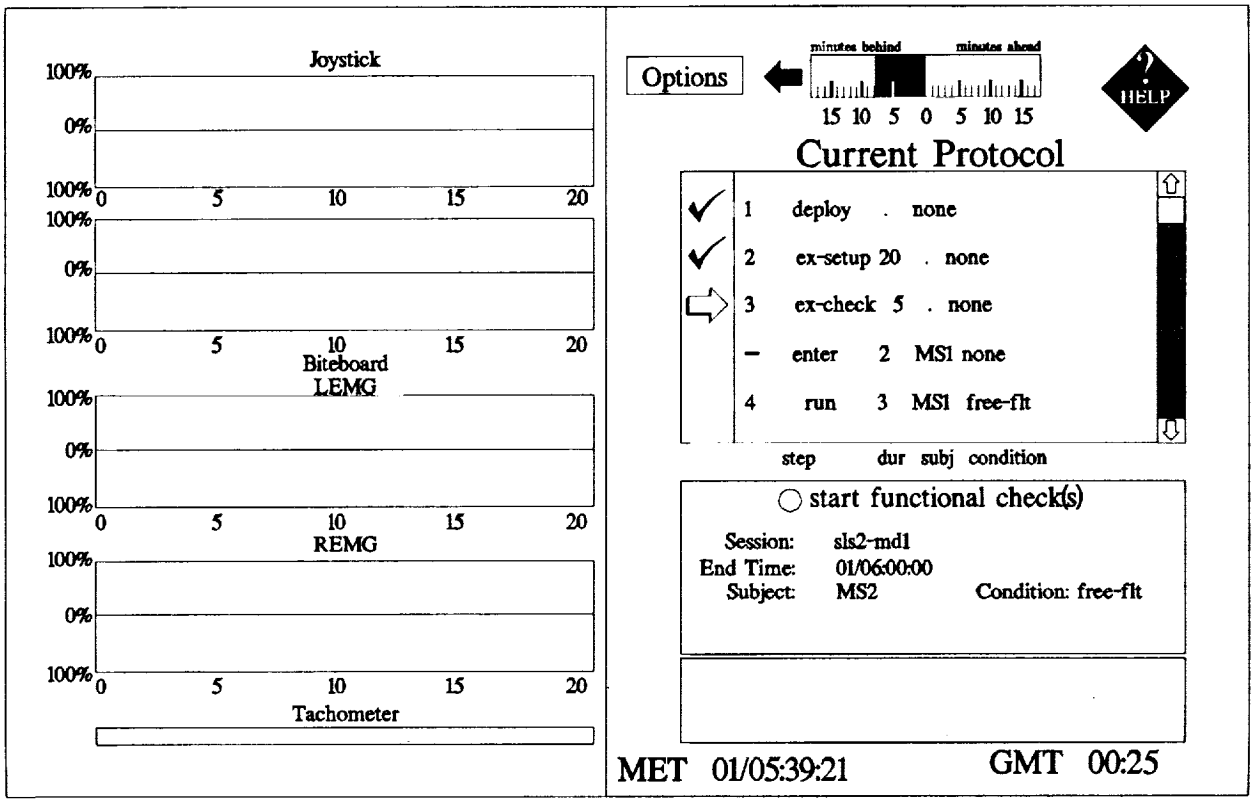


Figure 2

The third component of the system is the user interface, written in HyperCard (Claris Inc. and Apple Inc., both in Cupertino, California). The general interface (see Figure 2) consists of a vertically split screen, the left half of which is dedicated to graphic information while the right half contains active text and a graphic "delta clock". The delta clock shows the astronaut operators how well they are conforming with the experiment time-line. The active text consists of a scrolling script which reminds the operator of what steps have been completed and what remains to be done. Should the operator require more information concerning a step, clicking the mouse on the step text will produce a scrolling text box with detailed instructions for accomplishing the step.

Just before each dome data run, the operator "arms" [π] to look for initiation of dome rotation on the tachometer signal. Once rotation has been established data acquisition begins on all five channels at the rate of 225 samples per second. Should the tachometer signal be undetected for more than two seconds at the end of any trial, the run will be aborted by [π]. The usual cause for aborting a run is operator interruption by manually resetting the ECDS.

The data are shown in real-time on an oscilloscope-like display on the left side of the screen. At the beginning of each trial the data display from the previous trial is erased to be replaced by new data. Between trials the stored in separate files in unique directories (one for each run) and analysis is performed to be used to evaluate the run when all six trials are complete. While data acquisition is in progress all other activities are suspended. The delta clock shows a one dimensional horizontal bar graph indicating the number of minutes deviation from nominal progress. If the experiment protocol is on-time, the delta clock will be blank. Progress ahead of schedule is indicated by the bar growing to the right; if it falls behind the bar grows to the left. If the deviation is larger than eighteen minutes in either direction the bar cannot increase any further. Clicking the mouse on the delta clock causes the appearance of a dialog box informing the astronaut of the size of the deviation.

The system contains a very thorough trouble-shooting module. Failed functional checks can, with the operator's accord, lead directly to traversal of a malfunction tree. In the case a malfunction is beyond the astronaut's ability to repair, the system will attempt to continue the experiment without the affected signals. If the system cannot determine the cause of the malfunction from information it currently knows, it will ask the operator to perform pertinent test procedures. When a determination is made, a search for an appropriate repair procedure commences. Repair procedures are stored with the time necessary to perform them. The procedure execution time is compared with the delta clock to see if there is time available to do the repair. In addition, failures are scored by the severity of their impact on the data returned by the experiment. These two parameters allow the system to suggest carrying out the repair procedure or abandoning it to allow the experiment to continue in spite of the malfunction. The operator is given the opportunity to agree or overturn the suggested path. If the repair is undertaken, detailed directions are provided, including labeled exploded view drawings, tool locations, and step-by-step instructions. Unless the operator objects, at the end of any repair the affected signals are tested to recertify system status.

At any time when data acquisition is not active the operator can query [π] about alternative experiment protocols. [π] will generate two new protocols for any query. The "most desirable" protocol disregards all time constraints and attempts to optimize the scientific return based on all the observations so far collected for the entire mission. The "time optimized" protocol assumes the experiment will be required to terminate when specified by the mission time-line. It will try to optimize the scientific value of the data by omitting steps deemed less important. Determination of which steps to omit is based on the data collected during the mission. The astronaut may choose to execute either the "most desirable" or "time optimized" protocol or to stay with the current protocol.

[π] also possesses an efficient protocol editor which can be used to quickly create new experiment protocols or edit existing ones. The initial mission time-line and protocols were generated by the development team with reference to data published by NASA mission management. The likelihood that the time-line will change between stowage and launch is high, and the time-line is subject to change on-orbit. It is thus quite probable the astronauts will need to employ the editor and they have been trained to become very proficient with it. Any member of the crew can create a new, scientifically appropriate experiment protocol for the Rotating Dome experiment in less than two minutes. It is particularly hoped, should the mission be extended by one or more days, the availability of easily generated additional experiments may lead to greater scientific return from the experiment.

An important aspect of decisions concerning optimal scientific value is the "interestingness" of data generated by the subjects. This is certainly the most challenging and potentially rewarding issue in the whole program. Interestingness generally depends on what has been observed previously from a given subject. However, responses which are just "different" are not necessary interesting because there may be some obvious explanation for the difference. [π] is certainly not omniscient and much of what is obvious to the astronauts is not recognizable by the computer system. The best we can hope to do is flag what is believed may be interesting and allow the operator to agree with the observation, overturn it, comment about either action, or simply ignore the flag altogether.

A minimal text editor is provided to allow operators to log comments at any time when data acquisition is not in progress. Entries are automatically time stamped so they can be synchronized with experiment activities. It is not expected the astronauts will make much use of the text editor partly because it is somewhat difficult to type in zero-g: one tends to float away from the keyboard unless it is attached to one's body ([π] will be attached by velcro to a wall near the ECDS). In addition, the astronauts are provided with personal tape recorders on which they may voice comments. Unfortunately, the recordings are not time stamped and the tape recorders are often not operational. The crew is particularly aware of this problem and is willing to send their comments to us in real-time over shuttle voice communication when possible.

Conclusion

Our success getting [π] into space depended on a number of factors. The system is a *non-critical addition* to an already flight qualified experiment. Its size, mass, and power requirements are small. For a relatively small investment, it adds a number of valuable assets to the host experiment: a second data path to help guarantee capture of experiment critical data, the ability for the astronauts to see all the signals at once to monitor data quality, the capability to quickly assess changes in time-line to either take optimal advantage of extra time or to

minimize the damage caused by losing time, a dynamic script to remind the astronauts of the protocol and their progress through it, and a powerful trouble-shooting and repair assistant. It is designed, like a good servant, to speak only when asked and to offer quiet but effective help whenever it can.

We conclude that as technology allows more experiments of greater complexity to be packed into Spacelab (or the Space Station) while crew size remains unchanged, the desirability and value of having [π]-like systems to assist in experiment monitoring and control will increase. Applications to earth-bound domains appear to be abundant and valuable. Generalizing the technology to a broader range of scientific domains and the creation of powerful software tools to allow relatively easy generation of these systems should be well worth the investment.

Acknowledgments

Irv Statler of NASA Ames Research Center Human Factors Branch provided invaluable interface design advice. Bill Mayer, Ed Boughan, Jim O'Connor, and Fred Miller, all of the MIT Center for Space Research, made significant contributions to the [π] hardware and cut through endless red tape for us. Bob Renshaw and others at Payload Systems, Inc., Cambridge, MA., fabricated the AIB and helped with astronaut training. Rob Dye, John Hanks, John Fowler, and Don Powers, all with National Instruments, Inc., Austin, TX., furnished in depth LabVIEW support and advice. Jeff Shapiro, of RECOM, Inc., Moffett Field, CA., did Macintosh toolbox level system programming. Glenn Weinreb and Ken Kolias of GW Instruments, Sommerville, MA., supplied hardware help and advice about their A/D converter. Dr. Thomas Mayer of Apple Computer, Inc., Cupertino, CA., offered invaluable information and advice about the hardware and design of the PowerBook 170. Judy Stephenson, Jessica Sekerka, Gloria Salinas, Stuart Johnston and Terence Adams, all with Martin Marrietta Services, Inc., Houston, TX., assisted us at Johnson Space Center. Steve Cox of NASA Kennedy Space Center made sure our work there went smoothly. We would like to thank Michelle Zavada of the MIT Aeronautics Department for proofreading this text.

References

[YOU84] Young L. R., Perception of the body in space: mechanisms, Handbook of Physiology, The Nervous System III, Chapter 22, American Physiological Society, Smith I. D., ed., 1984.

[FRA93] Frainier R. J., Groleau N., Hazelton L. R., Colombano S. P., Compton M., Statler I., Szolovits P., and Young L. R., *PI-in-a-box: a knowledge-based system for space science experimentation,* Fifth Annual Conference on Innovative Applications of Artificial Intelligence, Washington D.C., July 1993

[GRO92] Groleau N., *Model-Based Scientific Discovery: A Study in Space Bioengineering*, unpublished Ph.D. Thesis , Dept. of Computer Science and Dept. of Civil and Environmental Engineering, M.I.T., Cambridge, Massachusetts, August 1992.

ENGINEERING DESIGN KNOWLEDGE RECYCLING in NEAR-REAL-TIME

Larry Leifer, Vinod Baya, and **George Toye***
Center for Design Research, Stanford University
560 Panama Street, Stanford, CA 94305-2232
leifer@sunrise.stanford.edu
baya@sunrise.stanford.edu
toye@sunrise.stanford.edu

Catherine Baudin and **Jody Gevins Underwood**
Artificial Intelligence Research Branch
NASA Ames Research Center MS-269/2, Moffett Field, CA 94035
baudin@ptolemy.arc.nasa.gov
gevins@ptolemy.arc.nasa.gov

ABSTRACT

It is hypothesized that the capture and reuse of machine readable design records is cost beneficial. This informal engineering notebook design knowledge can be used to model the artifact and the design process. Design rationale is, in part, preserved and available for examination. Redesign cycle time is significantly reduced (Baya et al, 1992). These factors contribute to making it less costly to capture and reuse knowledge than to recreate comparable knowledge (current practice). To test the hypothesis, we have focused on validation of the concept and tools in two "real design" projects this past year: 1) a short (8 month) turnaround project for NASA life science bioreactor researchers was done by a team of 3 Mechanical Engineering graduate students at Stanford University (in a class, ME210abc "Mechatronic Systems Design and Methodology", taught by one of the authors, Leifer; and 2) a long range (8 to 20 year) international consortium project for NASA's Space Science program (STEP: satellite test of the equivalence principle).

Design knowledge capture was supported this year by assigning the use of a Team-Design PowerBook. Design records were catalogued in near-real time. These records were used to qualitatively model the artifact design as it evolved. Dedal, an "intelligent librarian" developed at NASA-ARC, was used to navigate and retrieve captured knowledge for reuse.

INTRODUCTION

The Engineering Design Notebook (EDN®) concept has evolved rapidly. Whereas costly high performance workstations were used in the past to implement EDN, the concept has now been migrated to laptop PowerBooks (Appendix-B) for portability, ease of use and designer "ownership". Commercially available software (Appendix-C) replaced custom software used in our earlier research. In this form, the EDN concept of design knowledge capture was adopted by the SHARE project (see Appendix-A). In this context, EDN has been extended to support multiple, mobile and distributed design teams through use of the PowerBook as a notebook medium. The project is also informed by active collaboration with the ASK systems project, PACT (Palo Alto Collaborative Test-bed (for distributed design), and the NSF Synthesis Coalition (8 universities dealing with engineering knowledge capture and reuse in undergraduate education).

Deep engagement with real design activity, in parallel with the research program, has been a special feature of the NASA-Stanford collaboration on Generation and Conservation of Design Knowledge (GCDK). This strategy assures the utility of our results and keeps the research grounded in constant feedback from real designers. Last year, the FORD motor company sponsored "Continuously Variable Damper" a project in Stanford's Mechatronics Systems Design & Methodology course (ME210abc) was used post-facto to develop Dedal, an informal document navigation aid. This year we do real-time design knowledge capture and near-real-time reuse on the NASA-Bioreactor project. We have also captured "proof of concept" hardware development activity in the STEP project where rationale capture is particularly important during the proof-of-concept phase. This paper reports some of our findings in the ME210 test-bed environment.

ME210: a design research test-bed environment (Leifer, 1993)

The GCDK rapid prototyping environment is ME210abc (Figure-1), a 9-month long graduate engineering course in which teams of designers conceive, design and prototype substantial electromechanical systems for industrial

sponsors. The designers are typically first year graduate students with one to three years of industrial experience. They typically work in teams of three with coaching by faculty, staff and consultants from industry. The industry sponsored projects are usually multi-disciplinary, combining thermal, mechanical and electrical systems, sensors, actuators, and software. As in industry, the teams have tight deadlines, and must manage equipment and development budgets, engage in frequent design reviews, negotiate with the sponsors, vendors, and fabrication job-shops, etc.

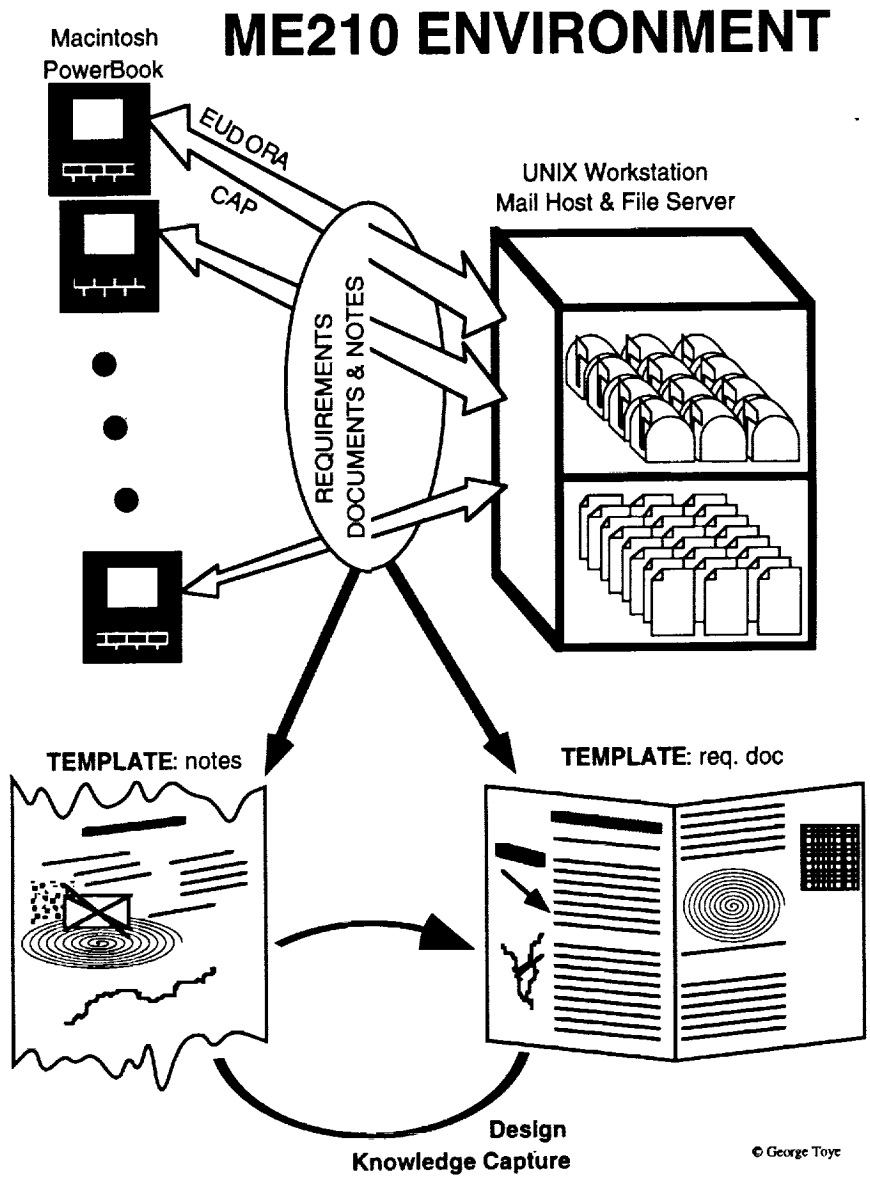


Figure 1: PowerBook mediated, network supported, capture and reuse of informal engineering knowledge.

The design process goes from requirements definition and conceptual design to a working prototype and final report in 9 months. The report, often running over 200 pages, is typically of more value to the sponsoring companies than the prototype because it not only documents the design, but also captures the students' experience, decision making process and knowledge relating to the project. Roughly 60% of the report consists of appendices of calculations, catalog pages and data sheets, test results, materials properties, contact logs and meeting notes, and a wealth of other information extracted or generated during the design. The remainder documents the decisions and rationale behind the final prototype as well as ideas pursued and ultimately abandoned.

The design process begins with extended negotiation and clarification with the sponsor about the design requirements and constraints. A requirements document is among the deliverables early in the course. However, as with all real design processes, the generation and clarification of requirements and constraints continues throughout the design. As design continues, information is continually gathered, sifted, sorted, and reorganized for presentation. The report is generated incrementally and submitted for review at the end of each academic quarter. Sections are reviewed more frequently as part of regularly scheduled design reviews. Each submission contains revised and reorganized information from the previous versions. As the teams are each working on different projects and not competing directly, there is more incentive to share knowledge than to hide it. Indeed, a class consensus rapidly develops concerning which tools, consultants, and handbooks are most helpful.

The engineering design course provides the GCDK project with important resources. First, it is a test-bed for evaluating tools, methodologies and concepts that we develop. The rapid turnover and aggressive design schedule allow us to obtain considerable empirical evidence over just a few years. The tight deadlines also ensure that designers will employ a new tool or method if and only if it is demonstrably helpful.

Second, the course provides a rich stream of design examples that cross multiple disciplines and levels of detail, from component design and fabrication to sub-assembly integration. Extensive interaction takes place not only among team members but also with other teams and with the "extended team" of sponsors, consultants, vendors, etc. The net result is a flood of information that must be captured and organized in near-real time. This coupling to real design activity provides a distinctive direction for tool development.

The course also provides many opportunities for observing and abetting design reuse. Often, sponsors come back with variations on a theme that appeared during previous years. For example, a client may ask for a packaging system one year and a materials handling and inspection system to go with it the next. Today, the main source of information about previous projects is a chronologically arranged library of final reports. Searching for relevant information is a tedious and inexact process. If a team needs information about precision assembly devices it is up to the faculty, staff and sponsors to recall which projects from which years are likely to yield something germane. Based on testimony by our industry sponsors, we know that this scenario is also typical on the job.

As part of GCDK, all delivered documents will be stored on CD ROMs and used to form an electronic design library for subsequent years. As we develop more tools for sorting, browsing, retrieving and querying the information, the electronic library will develop into the unified design representation and hyper web that we ultimately envision.

Share: a concurrent engineering vision (Toye et al, 1993)

Today's CAD systems do not adequately support the tasks on which engineers spend the most time: gathering and organizing information, communicating with clients, suppliers and colleagues, negotiating tradeoffs, and using each others' services. Engineers spend days or weeks locating catalog items, consultants, analysis tools, and production facilities. Often, they redo analyses and manufacturing plans because it is difficult to retrieve relevant examples from the past, or because the examples lack sufficient context or detail to adapt them to the circumstances. Sometimes, they forego analysis altogether because the cost of learning new tools exceeds their apparent worth. Then, when the parts are back from fabrication, it is all too often back to the drawing board because of an earlier failure to communicate some interface convention or constraint. The consequences of all these difficulties are well known, and include costly engineering change orders (ECOs), delays in procurement and fabrication of new prototypes, high reject rates, high maintenance costs and lost time to market.

To overcome these difficulties, we propose an open, network-oriented environment for concurrent engineering. The environment enables engineers to participate on a distributed team using their own tools and data bases. Specifically, it should provide:

- familiar displays that put useful information at the engineers' fingertips, including on-line notebooks, handbooks, requirements documents, and design libraries;
- collaboration services, including multimedia mail and desktop video conferencing, that enable team members to communicate and share tools and data;
- on-line catalog ordering and fabrication services, with information about pricing and shipping schedules and bid solicitation, leading to delivery of components without numerous phone calls to clarify the designers' intent;
- access to specialized network services for simulation, analysis and planning, (e.g., cost estimation, dynamics simulation) and access to shared engineering knowledge bases;

- a distributed product data management service that accepts postings from on-line tools and services, and maintains dependencies so that when changes occur, the right people are notified, the right tools invoked and the right sources consulted;
- an integration infrastructure that enables heterogeneous design tools and data bases to interoperate transparently across platforms, creating a shared project environment.

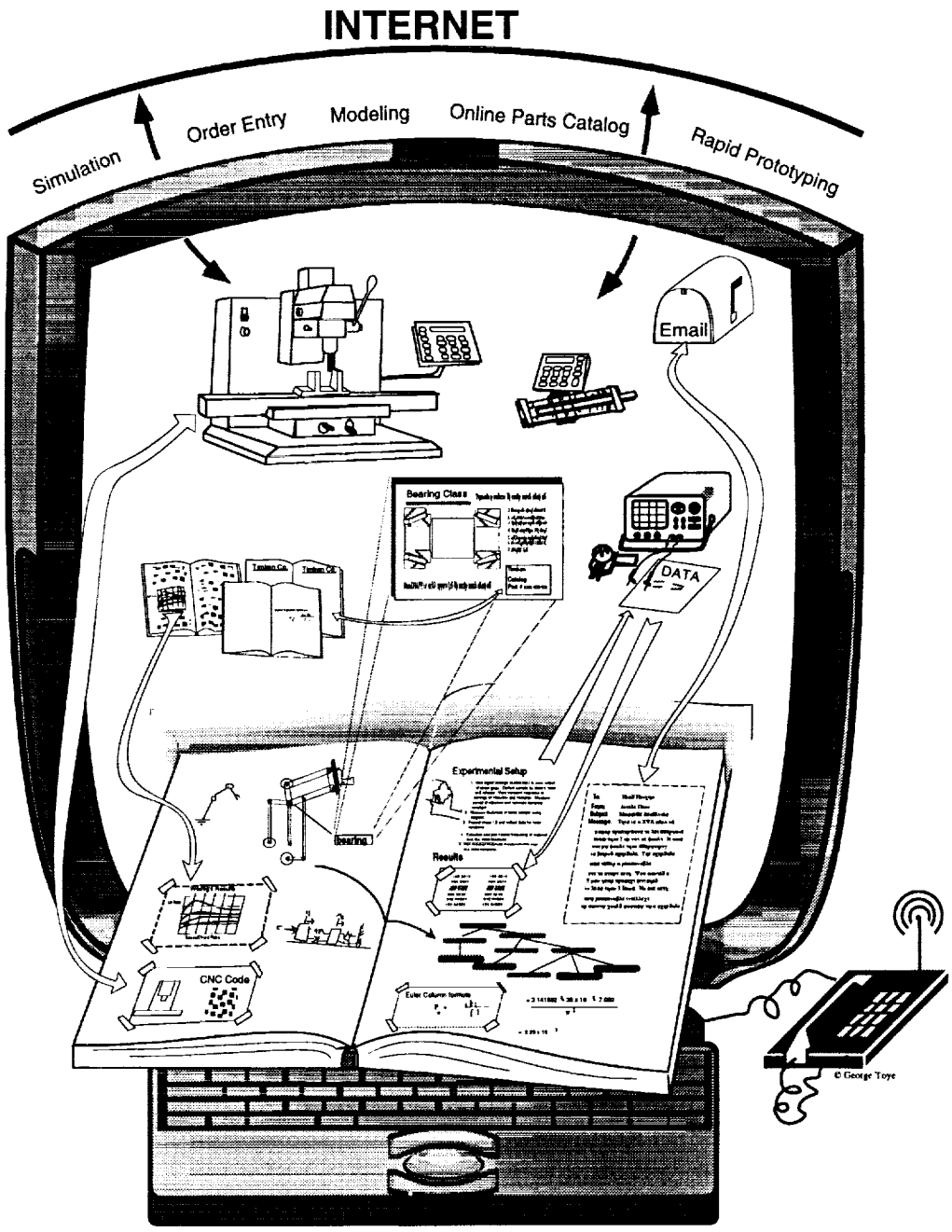


Figure 2: SHARE vision of the personal design notebook integrated into the world wide internet product development enterprise.

The windows on this world of networked resources and services will be multimedia "notebooks" in which to capture and organize information about a project: CAD drawings and solid models, audio notes, sketches, spreadsheets, pages from handbooks and catalogs, animated simulations, mail, excerpts from video conferences, and so forth. The goal is

a system that becomes one's preferred work environment for collaborating on everything from proposals to detailed designs.

What will life be like for an engineer on the Internet? He or she will browse on-line handbooks for relevant components or design models, and submit them to remote services for simulation or analysis. Interesting designs will be copied and pasted into the notebook, and annotated by adding hypertext links to related specifications, data, analysis tools, and components. These links represent constraints, rationale, and dependencies, some of which may point to entries in colleagues' notebooks. Notebook pages will be exchanged with colleagues and inserted into shared project notebooks. Users will navigate this distributed information web using browsing and search tools. Alternatively, they can invoke agents to keep track of dependencies and alert them (or other agents) to changes. Design conflicts that require negotiation will be resolved using multimedia e-mail or a notebook video conference. When a design is ready for fabrication, its specifications will be shipped over the network, perhaps through a broker, to the appropriate production, and procurement services.

Dedal: **an informal multi-media document navigator** (Baudin et al, 1993a)

Information retrieval systems that use conceptual indexing (Tong et al, 1989) to describe the information content perform better than syntactic indexing methods based on words from a text. However, since conceptual indices represent the semantics of a piece of information, it is difficult to extract them automatically from a document, and it is tedious to build them manually. We implemented an information retrieval system that acquires conceptual indices of text, graphics and videotaped documents. Our approach is to use an underlying model of the domain covered by the documents to constrain the user's queries. This facilitates question-based acquisition of conceptual indices: converting user queries into indices which accurately model the content of the documents, and can be reused. We discuss Dedal, a system that facilitates the indexing and retrieval of design documents in the mechanical engineering domain. A user formulates a query to the system, and if there is no corresponding index, Dedal uses the underlying domain model (Figure-3) and a set of retrieval heuristics to approximate the retrieval, and ask for confirmation from the user. If the user finds the retrieved information relevant, Dedal acquires a new index based on the query. We demonstrate the relevance and coverage of the acquired indices through experimentation.

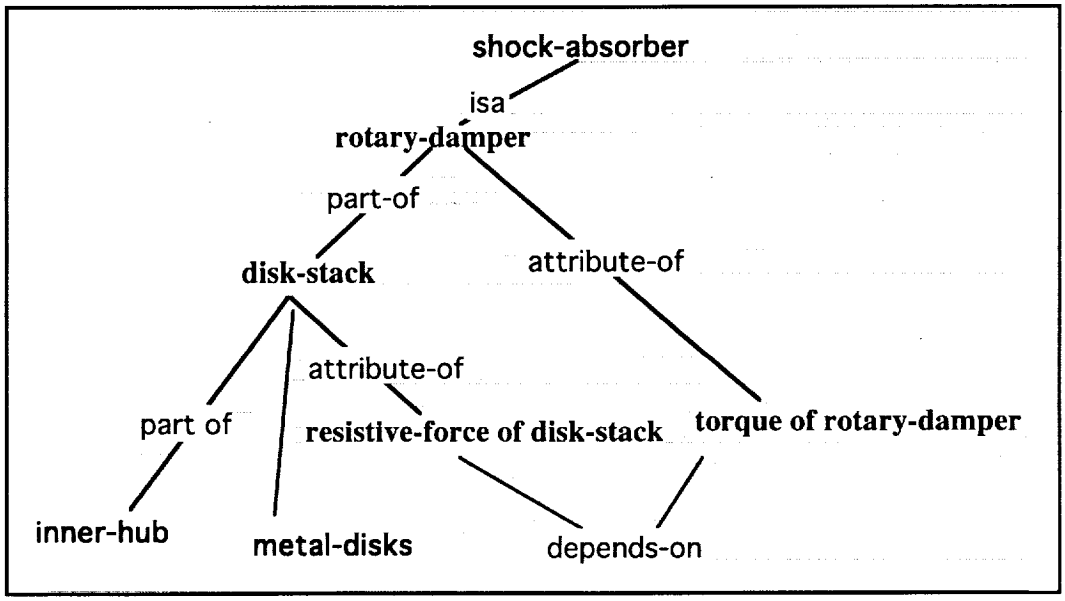


Figure 3: Objects and relations in the domain model

Our approach is to use a conceptual query language plus feedback from the user on the relevance of the documents retrieved in response to a query, to incrementally acquire new conceptual indices for that document. The user formulates a query to the system. If no document description exactly matches the query, the system approximates the retrieval and prompts the user for feedback on the relevance of the references retrieved. If a reference is confirmed, the query is turned into a new index. This extends relevance feedback techniques (Salton et al. 88) to the acquisition of conceptual indices.

This approach uses a question-based indexing paradigm (Osgood et al. 91)(Schank 91)(Mabogunje 93) where the query language and the indexing language have the same structure and use the same vocabulary. The assumption is

that the questions asked by users indicate the objects and relationships that are relevant to describe the content of the documents at a conceptual level appropriate for a class of users. However, in order to use the queries to acquire new indices the following conditions must be met by the query language:

Reusability: The query language must be general enough to create indices that will match a class of queries.

Relevance: The query language must be able to describe the information that the user is interested in articulating queries to acquire information in order to achieve a goal is in general a difficult task (Croft et al. 90). In our approach, the query formulation is constrained by a model of the domain covered by the documents and a model of the type of information designers are interested in.

Context independence: The query language must be able to generate indices that can be reused in different situations, for different users and different tasks.

The retrieval module takes a query from the user as input, matches the question to the set of conceptual indices and returns an ordered list of references related to the question. The retrieval proceeds in two steps: (1) exact match: find the indices that exactly match the query and return the associated list of references. If the exact match fails: (2) approximate match: activate the proximity retrieval heuristics.

Dedal currently uses fourteen proximity retrieval heuristics to find related answers to a question. For instance, segments described by concepts like "*decision* for lever material" and "*alternative* for lever material" are likely to be located in nearby regions of the documentation. The heuristics are described in detail in (Baudin et al. 92b).

Each retrieval step returns a list of references ordered according to a set of priority criteria. The user selects a reference and if the document is on line, goes to the corresponding segment of information (using the hypertext facility that supports the text and graphics documents). A user dissatisfied with the references retrieved can request more information and force Dedal to resume its search and retrieve other references.

CONCLUSIONS

Our work is based on the view of design as a process of gathering, organizing and exchanging information. The process begins with notes and concept sketches, catalog pages and evolves to encompass more formal representations as models are generated, tested and communicated to others and as the individual and group design records are annotated with decisions, revision notices, and fabrication orders. As the proportion of formal structured information increases, the ability of automated mechanisms to help people manage it also increases. However, even toward the end of a design, the proportion of informal to formal information remains disproportionately high.

Consequently, we focus our efforts on providing tools to help people capture and organize the multimedia e-mail messages, annotations and scratch work that make up the bulk of typical design records. These tools will have immediate applicability in the graduate design course that grounds the context of our work. We want to help design teams organize their personal notebooks and use the resulting documents for redesign during the term and in subsequent years. Our focus stems directly from our commitment to provide tools that are demonstrably useful in real design environments.

The graduate design course sequence provides us with a critical test-bed in this regard. It also provides a rich stream of information to capture and analyze. Our goal is to capture as much as feasible, on a continuing basis. In return, we will provide designers with automated retrieval, indexing and organization as these capabilities become available. We also acquaint designers with the potential of electronic design notebooks that serve both as personal and group design records and as windows to a world of services and resources on the Internet.

REFERENCES

Baudin, C., Gevins, J, Baya, V, 1993, "Using Device Models to Facilitate the Retrieval of Multimedia Design Information", Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'93), Chambrery, France, 1993a

Baudin, C., Gevins, J, Baya, V, Mabogunje, A., "Dedal: Using Domain Concepts to Index Engineering Design Information", Proceedings of the Meeting of the Cognitive Science Society, Bloomington, Indiana, 1992a

Baudin, C., Kedar, S., Gevins, J, Baya, V, "Question-based Acquisition of Conceptual Indices for Multimedia Design Documentation", Proceedings of the American Association for Artificial Intelligence (AAAI'93), Washington DC, 1993b

Baya, V, Gevins, J, Baudin, C, Mabogunje, A, Leifer, L., Toye, G., "An Experimental Study of Design Information Reuse", in proceedings of the 4th International Conference on Design Theory and Methodology, 1992.

Croft, W.B., Das, R., "Experiments with Query Acquisition and Use in Document Retrieval Systems". in Proceedings of SIGIR 1990.

EDN® is a registered trademark of the Performing Graphics Company

Lakin F., Wambaugh J., Leifer L., Cannon D., and Sivard C., "The Electronic Design Notebook: Performing Medium and Processing Medium", THE VISUAL COMPUTER, International Journal of Computer Graphics", Vol.5: 214-226, 1989

Lakin, F., Baya, V., Cannon, D. J., Brereton, M., Leifer, L., and Toye, G., "Mapping Design Information", Proceedings of the AAAI'92 Workshop on Design Rationale Capture and Use, San Jose, CA, July 1992

Leifer, L., "ME210abc: Mechatronics Systems Design & Methodology", an information package available from the Department of Mechanical Engineering Design Division, Stanford University, Stanford, CA 94305, June 1993

Mabogunje, A., "Harnessing Questions in Mechanical Design", Engineer: master of design in engineering Thesis, Stanford University, Department of Mechanical Engineering, July 1993

Osgood, R., Bareiss, R. "Question-based indexing", Technical report, The Institute for the Learning Sciences, Northwestern University, 1991

Salton, G., Buckley, C. "Improving Retrieval Performance by Relevance Feedback", Technical Report, Cornell University, 1988

Schank, R., Ferguson, W., Birnbaum, L., Barger, J., Greising, M., "ASK TOM: An Experimental Interface for Video Case Libraries" ILS technical report, March 1991.

Tong, M. R., Appelbaum, A., and Askman V. "A Knowledge Representation for Conceptual Information Retrieval", International Journal of Intelligent Systems, Vol. 4, 259-283, 1989

Toye, G., Cutkosky, M., Leifer, L., Tenenbaum, J., and Glicksman, J., "SHARE: a Methodology and Environment for Collaborative Product Development", Proceedings of the 2nd Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'93), CDR-TR #199330507, May, 1993

ACKNOWLEDGMENTS

Liming Lau, Ben Chou and Melissa Regan, the ME210 Bioreactor Design Team, worked diligently with the research team to make this study possible. They also made important contributions to our thinking and Dedal implementation. The value of user input to "participatory research" should not be under estimated. GCDK is supported in part by NASA contract NCC 2-514. SHARE is supported in part by the Department of Navy Contract N00014-92-J-1833. ME210 is supported in part by industry members of the Stanford Design Affiliates Program.

APPENDIX-A: Related work

The GCDK team is a unique component of the SHARE and PACT consortia on concurrent engineering. NASA's GCDK project has taken the leading position in regards to indexing, navigation and reuse of large, informal knowledge records. Of the following, it is the only project focused on NASA space science engineering applications.

SHARE (a scalable methodology and framework for concurrent engineering) is ARPA funded and jointly directed by Professors Leifer and Cutkosky in collaboration with Enterprise Integration Technologies Corporation (Dr. Jay Tenenbaum). PACT is a broad consortium of Palo Alto area laboratories. It includes the assembly laboratory lead by Prof. Latombe in computer science at Stanford; the logic Design World project supported by Hewlett-Packard and lead by Prof. Genesereth at Stanford; the How Things Works project lead by Dr. Tom Gruber and Professor Richard Fikes at the Stanford Knowledge Systems Laboratory; the NVisage program at Lockheed Corporation's AI laboratory lead by Dr. Bill Mark; and members of the Stanford Manufacturing Systems Engineering Program (MSE). Professor Leifer is also co-PI with Professor Sheri Sheppard on the NSF sponsored National Engineering Education Synthesis Coalition. Consortia members include: Cornell University, Hampton University, Tuskegee University, Southern University, Cal Poly San Luis Obispo, University of California at Berkeley, Iowa State University and Stanford

University. These collateral associations promote widespread dissemination of NASA sponsored work and assure that the work itself is well informed by the activity of others.

APPENDIX-B: EDN laptop computer hardware configuration

Apple Computer PowerBook 160 (adequate may go to Duos next year)
8 MB memory (minimum)
80 MB hard disk drive (adequate with file server backups)
Global Village Silver PowerPort fax-modem (adequate)

The design environment included some additional hardware, including: 4 Hewlett-Packard CAD stations, 4 Macintosh IIci workstations, 1 Microsoft DOS compatible Intel PC. In addition to computer hardware, a full complement of rapid physical prototyping equipment was available.

APPENDIX-C: EDN laptop computer software configuration

FrameMaker 3.0: a document processor
Microsoft Excel 4.0: a spreadsheet processor
Aldus Persuasion 2.0: a presentation environment
Eudora 1.3: a public domain electronic mail manager
AOS System 7.0: the Apple Macintosh operating system

A variety of engineering software tools were available in the laboratory on desktop workstations: e.g., CAD, CAM, CAE, Symbolic Math Modeling. The environment was networked using Local Talk (to be upgraded next year to ethernet) and supported by an file server on the Internet for file backup and electronic mail.

N94-34062

A Toolbox and a Record for Scientific Model Development

Thomas Ellman
Department of Computer Science
Hill Center for Mathematical Sciences
Rutgers University, New Brunswick, New Jersey 08903
ellman@cs.rutgers.edu

Abstract

Scientific computation can benefit from software tools that facilitate construction of computational models, control the application of models, and aid in revising models to handle new situations. Existing environments for scientific programming provide only limited means of handling these tasks. This paper describes a two pronged approach for handling these tasks: (1) designing a "Model Development Toolbox" that includes a basic set of model constructing operations; (2) designing a "Model Development Record" that is automatically generated during model construction. The record is subsequently exploited by tools that control the application of scientific models and revise models to handle new situations. Our two pronged approach is motivated by our belief that the model development toolbox and record should be highly interdependent. In particular, a suitable model development record can be constructed only when models are developed using a well defined set of operations. We expect this research to facilitate rapid development of new scientific computational models, to help ensure appropriate use of such models and to facilitate sharing of such models among working computational scientists. We are testing this approach by extending SIGMA, an existing knowledge-based scientific software design tool.

1 Problem: Support for Construction, Testing, Application and Revision of Scientific Models

Computational science presents a host of challenges for the field of knowledge-based software design. Scientific computation models are difficult to construct. Models constructed by one scientist are easily mis-applied by other scientists to problems for which they are not well-suited. Finally, models constructed by one scientist are difficult for others to modify or extend to handle new types of problems. Existing knowledge-based scientific software design tools, such as SIGMA [Keller and Rimon, 1992], provide only limited means of overcoming these difficulties. For example, SIGMA facilitates model construction by providing scientists with high-level data-flow language for expressing models in domain-specific terms. Although

SIGMA represents an advance over conventional methods of scientific programming, it supports only certain aspects of the model development process. In particular, SIGMA focuses mainly on automating the process of assembling equations and compiling them into an executable program. Construction of scientific models actually involves much more than the mechanics of building a single computational model. In the course of developing a model, a scientist will often test a candidate model against experimental data or against a priori expectations. Test results often lead to revisions of the model and a consequent need for additional testing. During a single model development session, a scientist typically examines a whole series of alternative models, each using different simplifying assumptions or modeling techniques. A useful scientific software design tool must support these aspects of the model development process as well. In particular, it should propose and carry out tests of candidate models. It should analyze test results and identify models and parts of models that must be changed. It should determine what types of changes can potentially cure a given negative test result. It should organize candidate models, test data and test results into a coherent record of the development process. Finally, it should exploit the development record for two purposes: (1) automatically determining the applicability of a scientific model to a given problem; (2) supporting revision of a scientific model to handle a new type of problem. Existing knowledge-based software design tools must be extended in order to provide these facilities.

2 Solution: A Model Development Toolbox and Record

We plan to attack this problem using two related ideas: First, we will define a "Model Development Toolbox". The toolbox will define a set of generic model development steps that are taken by most scientists in the course of developing scientific computational models. The envisioned generic steps include: (1) mapping equations onto physical situations; (2) fitting models against experimental data; (3) sanity checking model outputs against a priori sign, monotonicity or order of magnitude expectations; (4) testing models against experimental data; (5) analysis of test results; and (6) modification of models in response to test results. We plan to implement this toolbox in a scientific model development environment that guides scientist-users through the model development process. Second, we plan to design a "Model Development Record". The record will contain machine readable documentation of the entire model development process. To begin with, the record should describe the goals the model is intended to fulfill. For example, this might include a representation of the questions the model is (and is not) intended to answer The record should also describe the sequence of candidate models that were constructed in the course of developing the final model. For each candidate model, the record should describe: (1) the model itself; (i.e., equations and dataflow graphs), (2) assumptions underlying the model; (3) fitting techniques used to instantiate free parameters of the model; (4) sanity checks that were performed; and (5) tests against empirical data that were performed. The record should also describe (6) the temporal sequence of candidate models as well as (7) logical dependencies between test results on early models and modeling choices made in constructing subsequent, more refined models.

Tools for checking applicability of scientific models to new problems will rely heavily on the model development record. Important applicability checks include: determining

whether a proposed use of a model is consistent with the goals the model was originally intended to fulfill; determining if a new problem lies within the range of input parameter values for which the model was tested; and testing assumptions underlying the equations that were incorporated into the model. Each of these checks requires access to various aspects of the model development record. Likewise, tools that support model revision will also rely heavily on the model development record. Important types of model revision include: extending/modifying the model to handle a wider/different range of input parameters; re-fitting free parameters of the model to new empirical data; changing the assumptions used to model a physical process; adding/deleting physical processes to/from the model; and changing the overall purpose of the model. A model revision tool should automatically determine when a revision is needed (e.g., by determining that a new problem falls outside the range of problems handled by the original model, or by detecting discrepancies between empirical data and outputs of the model). It should suggest changes to the model that have the potential to cure the problem (e.g., by reasoning about sensitivities of outputs with respect to changes in intermediate results, or by reasoning about the effects of potential changes in assumptions on the outputs of the model). Finally the system should assist in re-validating the new model, (e.g., by suggesting new tests of validity, and carrying out and evaluating such tests.) In many cases, models may be revised by "replaying" a portion of the development record that led to the original model. Replay will require access to logical dependencies among test results and modeling choices found in the development record, using techniques similar to derivational analogy [Mostow, 1989] and transformational implementation [Balzer, 1985].

3 Model Development System Architecture

The overall architecture of our envisioned system is shown in Figure 1. The model development toolbox will serve as a front end to the whole system. The toolbox can interact with a human user to build an initial model in some scientific domain. It can also interact with a user in order to revise an existing model to handle a new situation. Finally, the toolbox also includes facilities for controlling the application of scientific models. As the toolbox guides the user through a series of model building, testing and revision steps, it interacts with several data bases. The model fragment data base contains the basic building blocks of scientific models. The toolbox uses techniques embodied in the SIGMA system to combine model fragments into one or more "current working models". As working models are constructed, they are tested against test data drawn from a test data base. Likewise, as tests are run, results are incorporated back into the test data base. As the initial model development process unfolds, the toolbox leaves a structured trace of the process in the model development record. When operating in replay mode, the toolbox is guided by a model development record constructed previously. Some portions of our system have already been implemented in SIGMA: These include the model fragment data base, the test data base and a framework for representing working models. Nevertheless, we expect that the representations used in SIGMA for these modules will need to be enhanced. A rudimentary version of the toolbox has also been implemented in SIGMA; however, most of our toolbox remains to be designed and build. The model development record is entirely new.

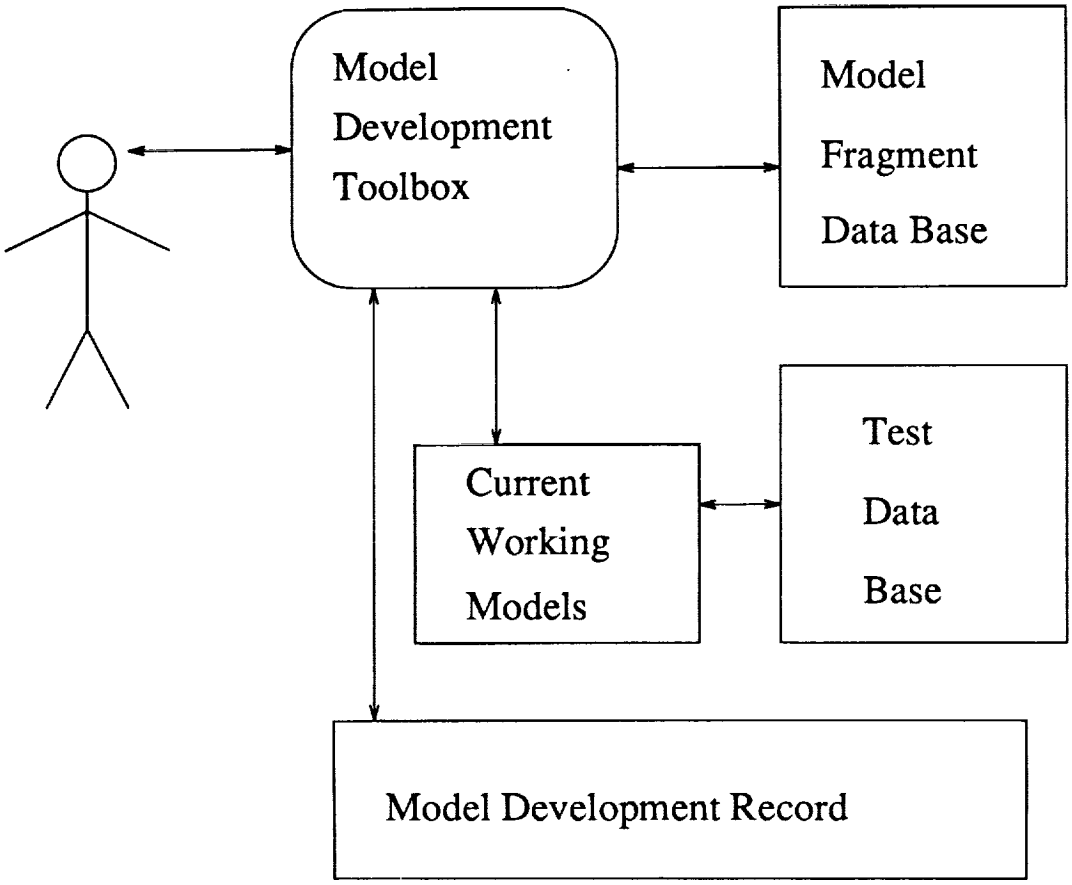


Figure 1: Model Development System Architecture

4 An Illustrative Example

As an illustration of the envisioned system, consider the following example of building a scientific model of the atmosphere of Saturn's moon Titan[1] The model takes as input a set of measurements of the refractivity of the atmosphere at various altitudes. The model is intended to compute atmospheric temperature and pressure at these altitudes. As the toolbox guides the human scientist through the model building process, it presents him with various modeling choices. For example he must decide which gases are to be included in the model. Let's suppose he chooses to include methane and nitrogen. He must also choose whether to use the ideal gas law, or a non-ideal gas law, to compute temperature from density and pressure. Let's suppose he chooses to use the ideal gas law. As the model is built, the user might declare certain expected properties of the output, e.g., that temperature and pressure are both positive numbers and are monotonically decreasing functions of altitude. The toolbox records these expectations in the model description in a representation that allows them to be checked automatically.

Once a preliminary model is constructed, the user may test the model on any available test

[1]The example is taken from [Keller and Rimon, 1992] and slightly modified. The details of example are not intended to be entirely accurate from the standpoint of atmospheric modeling.

data sets. If only input test data is available, (i.e., refractivity measurements) the system simply verifies that the outputs conform to declared expectations (i.e., the temperature and pressure are monotonically decreasing positive functions). If previously known output data is available, the system compares the known data to the outputs of the model and informs the user of discrepancies. For example, such tests might indicate that the pressure predictions are two low. The system might then suggest that the low pressure problem can be cured by either a change in the identities of the component gases, or by an addition of new gases into the mixture. Let's suppose the user decides to add ammonia into the mixture of gases. The system would revise the original model to include ammonia. It would also store the old model in the development record, along with a summary of the successful and unsuccessful tests performed on it. The cycle of model construction, testing and revision might be repeated several times before the user decides the model is satisfactory. The resulting model development record would include a description of the final model along with all the models examined along the way.

Once a satisfactory model is constructed by a human scientist, the model might be borrowed by a scientist working on a related problem, e.g., someone modeling the atmosphere of another satellite. The toolbox would guide such a new user through a series of steps designed to modify and validate the model for the new application. The system would examine the original model development record to determine what tests were performed on the original model. It would attempt to carry out analogous tests in the new setting. For example, the system might determine that, in the new setting, the model generates temperature or pressure levels for which the ideal gas law is not valid. The system would inform the user of the problem and suggest possible changes, e.g., using a non-ideal gas law, or changing the identities of gases in the mixture. Once the user chooses among the suggested revisions, the system would modify the model, update the record, and repeat any previous tests whose results are no longer valid. The cycle would repeat until the model passes all the tests suggested by the system and the user.

5 Key Research Issues

5.1 Model Development Toolbox Issues

A number of important research issues must be addressed along the way to implementing the model development architecture described in Figure 1. Implementation of the model development toolbox requires identifying a set of generic model building steps, and constraining the flow of control among them. Furthermore, in order that the toolbox support revision of scientific models, a number of distinct inference tasks must be performed. We thus expect to address the following questions in the course of designing the model development toolbox:

- *What primitive operations appear during the course of model development and model revision?* Potential primitives include: Select a model fragment to be used to compute a quantity. Replace one model fragment with another from the same class; Instantiate a generic model fragment in a specific scenario; Fit free parameters of a model against test data; Run a model on a set of test data; Compare test results to expected results; Add or remove a datum from the set of inputs or outputs of a model; Change the dimensionality of the inputs or outputs of a model.

- *What regularities appear in the sequences of operations that occur during model development and revision?* For example: Many models are hierarchically structured, i.e., they contain sub-models and sub-sub-models, etc. Potential construction strategies include: Top-down (breadth-first) and bottom-up (depth-first) or some combination. For each sub-model, the following sequence sequence of operations may be invoked: Select a model fragment incorporating suitable approximations; Run the model on a set of test data; Evaluate the test results; Revise the model fragment selection; Repeat, etc.

- *How can a system automatically detect circumstances in which a model must be revised?* For example: Input data can be compared to range constraints identified through previous tests; Output data can be checked for the expected sign, monotonicity or order of magnitude, when such expectations have been previously associated with the model; Outputs or intermediate results can be tested for consistency with simplifying assumptions; Outputs can be tested against benchmark data sets.

- *How can a system automatically determine which modeling choices must be revised to cure an identified problem?* A number of previously developed techniques may be applicable when suitably extended: For example, model selection methods that reason about the impact of choices on the sign of the error of a model's output are reported in [Addanki *et al.*, 1991] and [Weld, 1991]. Model selection methods that reason about the order of magnitude of the error may be developed by extending the techniques reported in [Raiman, 1991] and [Williams, 1991]. Likewise, model-selection methods relying on absolute error estimates may also be useful [Ellman *et al.*, 1993], [Falkenhainer, 1993] Furthermore, new techniques may be needed in order to reason about consistency between modeling choices in separate sub-models of a single larger model. Finally, truth-maintenance methods will likely prove useful in this portion of the system [De Kleer, 1986].

5.2 Model Development Record Issues

In order to design a model development record, we must identify the types of information that need to be included in the record, as well as suitable means of representing and organizing such information. The content of the record must be determined largely by the requirements of the processes the record is intended to support, i.e., developing models, controlling applicability of models and revising models. We thus expect to address the following questions in the course of designing the model development record:

- *What information about the goals of a scientific model must be represented in order to support development, application and revision of scientific models?* Potentially relevant information includes: A representation of the questions the model is intended to answer; A description of the quantities or relationships the models is (and is not) designed to compute; Desired accuracy levels; Legitimate and illegitimate uses of the outputs of the model.

- *What information about individual models and model fragments should be represented?* Aside from the models themselves, potentially relevant information includes: Restric-

tions on the input data; Testable simplifying assumptions that justify the approximations used in the model; Expectations regarding the sign, monotonicity or order of magnitude of the outputs or intermediate results.

- *What information about tests and test data should be represented?* Potentially relevant information includes: The purpose of the test; The model and test data used; Analyses performed on the test output data; Indications of satisfied and unsatisfied expectations.
- *How should the whole model development record be organized?* The record should include both the sequence of operations that led to the final model, as well as the development paths that failed and resulted in backtracking to earlier decision points. Thus the record needs to represent both temporal and logical relationships between different parts of the record.
- *What types of logical relationships between different parts of the record should be recorded?* Potentially relevant data includes: Dependencies between modeling choices in different parts of the model; Dependencies between goals and tests; Dependencies between test results and subsequent decisions.

We are pursuing this research by building an extension to the SIGMA system [Keller and Rimon, 1992] currently being developed at NASA Ames. We plan to develop the system by rationally reconstructing the process of developing and revising one of the two scientific models already implemented in SIGMA: a model of the atmosphere of Titan [McKay *et al.*, 1989], or a model of forest ecosystem processes [Running and Coughlan, 1988]. Additional candidate testbed domains include racing yacht design and jet engine nozzle design, each of which we have used as testbed applications for our previous work in the area of artificial-intelligence and computer-aided design [Ellman *et al.*, 1993].

6 Summary

The model development toolbox and record is expected to support a variety of activities that occur in the course of developing scientific computation models. These activities include construction and testing of new models; controlled application of models to specific problems, and revision of models to handle new situations. The system is also expected to promote rapid development of new scientific computational models, more reliable use of scientific models among computational scientists; wider sharing of scientific models within communities of scientists; and deeper understanding among scientists of the assumptions and modeling techniques incorporated in the models they use.

7 Acknowledgements

The research presented in this document has benefited from discussions with Richard Keller, Saul Amarel, Haym Hirsh, Lou Steinberg, Andrew Gelsey, John Keane and Mark Schwabacher.

References

[Addanki *et al.*, 1991] S. Addanki, R. Cremonini, and J. Scot. Graphs of models. *Artificial Intelligence*, 50, 1991.

[Balzer, 1985] R. Balzer. A 15 year perspective on automatic programming. *IEEE Transactions on Software Engineering*, SE-11(11):1257–1268, November 1985.

[De Kleer, 1986] J. De Kleer. An assumption-based tms. *Artificial Intelligence*, 28:127 – 162, 1986.

[Ellman *et al.*, 1993] T. Ellman, J. Keane, and M. Schwabacher. Intelligent model selection for hillclimbing search in computer-aided design. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, D.C., 1993.

[Falkenhainer, 1993] B. Falkenhainer. Ideal physical systems. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, D.C., 1993.

[Keller and Rimon, 1992] R. Keller and M. Rimon. A knowledge-based software development environment for scientific model-building. In *Proceedings of the Seventh Knowledge-Based Software-Engineering Conference*, Tysons Corner, VA, 1992.

[McKay *et al.*, 1989] C. McKay, J. Pollack, and R. Courtin. The thermal structure of titan's atomsphere. *Icarus*, 80:23 – 53, 1989.

[Mostow, 1989] J. Mostow. Design by derivational analogy: Issues in the automated replay of design plans. *Artificial Intelligence*, 40:119 – 184, 1989.

[Raiman, 1991] O. Raiman. Order of magnitude reasoning. *Artificial Intelligence*, 50, 1991.

[Running and Coughlan, 1988] S. Running and J. Coughlan. A general model of forest ecosystem processes for regional applications. *Ecological Modeling*, 42:125 – 154, 1988.

[Weld, 1991] D. Weld. Reasoning about model accuracy. Technical Report 91-05-02, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 1991.

[Williams, 1991] B. Williams. A theory of interactions: Unifiying qualitative and quantitative algebraic reasoning. *Artificial Intelligence*, 50, 1991.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1994	Conference Publication

4. TITLE AND SUBTITLE
Seventh Annual Workshop on Space Operations Applications and Research (SOAR '93) – Volumes I and II

5. FUNDING NUMBERS

6. AUTHOR(S)
Kumar Krishen, Editor

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lyndon B. Johnson Space Center
Houston, TX 77058

8. PERFORMING ORGANIZATION REPORT NUMBER
S-749

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546
U.S. Air Force, Washington, D.C. 23304

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA CP 3240

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Available from the
NASA Center for Aerospace Information
800 Elkridge Landing Road
Linthicum Heights, MD 21090
Subject Category: 99

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document contains papers presented at the Space Operations, Applications and Research Symposium (SOAR) Symposium hosted by NASA/Johnson Space Center (JSC) on August 3-5, 1993, and held at JSC Gilruth Recreation Center. The symposium was cosponsored by NASA/JSC and U.S. Air Force Materiel Command. SOAR included NASA and USAF programmatic overviews, plenary session, panel discussions, panel sessions, and exhibits. It invited technical papers in support of U.S. Army, U.S. Navy, Department of Energy, NASA, and USAF programs in the following areas: robotics and telepresence, automation and intelligent systems, human factors, life support, and space maintenance and servicing. SOAR was concerned with Government-sponsored research and development relevant to aerospace operations. More than 100 technical papers, 17 exhibits, a plenary session, several panel discussions, and several keynote speeches were included in SOAR '93. These proceedings, along with comments and suggestions made by panelists and keynote speakers, will be used to assess progress made in joint USAF/NASA projects and activities to identify future collaborative/joint programs. SOAR '93 was the responsibility of the USAF NASA Space Technology Interdependency Group Operations Committee. Symposium proceedings include papers presented by experts from NASA, the USAF, USA, and USN, U.S. Department of Energy, universities, and industry.

14. SUBJECT TERMS
Navigation; machine perception and exploration; ground operations teams; space physiology; operations challenges; artificial intelligence; robotics and telepresence research challenges; enhanced environments; medical operations; psychophysiology

15. NUMBER OF PAGES
839

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited